The Logical View on Continuous Petri Nets

MICHAEL BLONDIN, DIRO, Université de Montréal, Canada,
LSV, CNRS & ENS Cachan, Université Paris-Saclay, France
ALAIN FINKEL, LSV, CNRS & ENS Cachan, Université Paris-Saclay, France
CHRISTOPH HAASE, LSV, CNRS & ENS Cachan, Université Paris-Saclay, France
SERGE HADDAD, LSV, CNRS & ENS Cachan, Université Paris-Saclay & INRIA, France

Continuous Petri nets are a relaxation of classical discrete Petri nets in which transitions can be fired a fractional number of times, and consequently places may contain a fractional number of tokens. Such continuous Petri nets are an appealing object to study since they over approximate the set of reachable configurations of their discrete counterparts, and their reachability problem is known to be decidable in polynomial time. The starting point of this paper is to show that the reachability relation for continuous Petri nets is definable by a sentence of linear size in the existential theory of the rationals with addition and order. Using this characterization, we obtain decidability and complexity results for a number of classical decision problems for continuous Petri nets. In particular, we settle the open problem about the precise complexity of reachability set inclusion. Finally, we show how continuous Petri nets can be incorporated inside the classical backward coverability algorithm for discrete Petri nets as a pruning heuristic in order to tackle the symbolic state explosion problem. The cornerstone of the approach we present is that our logical characterization enables us to leverage the power of modern SMT-solvers in order to yield a highly performant and robust decision procedure for coverability in Petri nets. We demonstrate the applicability of our approach on a set of standard benchmarks from the literature.

CCS Concepts: •**Theory of computation** → **Automata over infinite objects; Logic and verification; Parallel computing models;**

Additional Key Words and Phrases: Petri nets, vector addition systems, coverability, arithmetic theories, linear programming

ACM Reference Format:
Michael Blondin, Alain Finkel, Christoph Haase, Serge Haddad, 2016. The Logical View on Continuous Petri Nets. *ACM Trans. Comput. Logic* V, N, Article A (January YYYY), 28 pages.
DOI: 0000001.0000001

1. INTRODUCTION

Petri nets are a well-established mathematical model for modeling and reasoning about distributed and concurrent infinite-state systems. They provide a high level of abstraction that allows for employing them in a great variety of application domains, ranging, for instance, from formal verification of concurrent programs to the modeling

M. Blondin was supported by the Fonds de recherche du Québec – Nature et technologies (FRQNT), by the French Centre national de la recherche scientifique (CNRS), and by the "Chaire Digiteo, ENS Cachan — École Polytechnique". C. Haase was supported by Labex Digicosme, Univ. Paris-Saclay, project VERICONISS. S. Haddad was supported by the ERC project EQualIS (FP7-308087).
Author's addresses: M. Blondin, Institut für Informatik, Technische Universität München, Boltzmannstr. 3, 85748, Munich, Germany, email: blondin@in.tum.de; A. Finkel and S. Haddad, LSV, CNRS & ENS Cachan, 61 avenue du Président Wilson, 94235 Cachan Cedex, France, email: {finkel, haddad}@lsv.ens-cachan.fr; C. Haase, Department of Computer Science, University of Oxford, Parks Rd, Oxford OX1 3QD, United Kingdom, email: christoph.haase@cs.ox.ac.uk.

DOI: 0000001.0000001

of biological, chemical and business processes (see e.g. [German and Sistla 1992; Ball et al. 2001; Reddy et al. 1996; Heiner et al. 2008; van der Aalst 1998]).

The *reachability problem* is the central decision problem for Petri nets: given an initial and a target configuration, the reachability problem asks whether there exist a sequence of transitions leading from the initial configuration to the target configuration. Even though the reachability problem has been shown decidable several times over the course of the last 35 years [Mayr 1981; Kosaraju 1982; Lambert 1992; Leroux 2009; Leroux 2011; Leroux 2012], no practically useful algorithm has yet been found. In particular, the problem is EXPSPACE-hard [Lipton 1976], and the best known upper bound is $\mathbf{F}_{\omega^3}$, "cubic Ackermannian" time [Leroux and Schmitz 2015].

In order to alleviate these high computational costs, one approach is to find suitable over approximations of the sets of reachable configurations. If a configuration is not included in an over approximation, it is guaranteed not to be reachable in the standard semantics. For instance, one may relax the semantics of Petri nets by allowing places to contain a negative amount of tokens along a run before ending in a configuration in which all places have a non-negative number of tokens. Then testing for reachability simply amounts to solving a system of linear Diophantine equations, which can be achieved efficiently, both in terms of computational complexity (the problem is NP-complete [Borosh and Treybing 1976]) and in practice via IP or SMT solvers. Such an approach has, for instance, been investigated by Esparza and Melzer [2000] and Esparza et al. [2014], where state equations and so-called trap constraints are used in a semi-decision procedure for disproving reachability.

The focus of this paper is on *continuous Petri nets*, a variant of Petri nets in which a transition may be fired a fractional number of times, and where places may consequently carry a non-negative *fractional* number of tokens. Continuous Petri nets were introduced by David and Alla [1987] and provide a good over approximation of standard Petri nets, see e.g. [Recalde et al. 1999; David and Alla 2010]. Moreover, their big advantage is that their reachability problem has recently been shown decidable in polynomial time [Fraca and Haddad 2015], contrasting greatly with the EXPSPACE-hardness of their discrete counterparts.

1.1. Our Contribution

The starting point of this paper is to show that the reachability relation of continuous Petri nets is definable by a sentence of linear size in the existential theory of the rationals with addition and order, $\mathrm{FO}\langle\mathbb{Q},+,<\rangle$. In contrast to discrete Petri nets, this in particular yields an easy proof of the fact that reachability sets of continuous Petri nets are closed under all Boolean operations and projections. It moreover entails decidability and complexity results for decision problems of continuous Petri nets that are definable in $\mathrm{FO}\langle\mathbb{Q},+,<\rangle$, albeit not necessarily with optimal upper complexity bounds. The decision problems we consider in this paper are reachability set inclusion, ε-liveness and home state problems, and we can show for many of them tight upper bounds. In particular, we settle the precise complexity of the inclusion problem, which was left open by Fraca and Haddad [2015] and Blondin et al. [2016].

A further main contribution of our work is to derive from our logical characterization of reachability in continuous Petri nets a highly performant and robust decision algorithm for the coverability problem for *discrete* Petri nets. The coverability problem is a relaxation of the reachability problem and defined as follows: given an initial and a target configuration, does there exist a sequence of transitions leading from the initial configuration to a configuration that *covers* the target configuration, i.e., one that is *larger or equal* to the target configuration? This problem was one of the first problems for Petri nets shown decidable [Karp and Miller 1967], and is known to be EXPSPACE-complete [Lipton 1976; Rackoff 1978]. It has attracted much attention in

the literature since it enables the verification of safety properties of many systems while being algorithmically and empirically easier to solve than the reachability problem. The *backward algorithm* [Arnold and Latteux 1978; Abdulla et al. 1996] is one of the most prominent algorithms for deciding coverability. Starting from the target configuration, it successively computes a set of *minimal basis elements* which provide a finite symbolic representation of the set of configurations starting from which the target configuration can be covered. The main bottleneck of the algorithm is that the size of the finite representation may grow doubly exponentially during the execution of the backward algorithm [Bozzelli and Ganty 2011]. This problem is commonly known as the symbolic state explosion problem [Delzanno et al. 2004].

In this paper, we revisit the classical backward algorithm for the coverability problem and use reachability in continuous Petri nets as a pruning heuristic in order to keep the set of minimal basis elements small. Since continuous Petri nets over approximate the reachability set of their discrete counter parts, if no configuration in the set of configurations defined by a minimal basis element is continuously reachable then no configuration is discretely reachable either. Therefore, minimal basis elements defining a set of configurations that are not continuously reachable from the initial configuration can be discarded during the execution of the algorithm, keeping the set of minimal basis elements small and thus speeding up the algorithm. In particular, our logical characterization enables us to leverage the power of modern SMT solvers in order to decide continuous reachability efficiently. The usefulness of our approach is demonstrated by evaluating it on a set of standard benchmarks from the literature. We show that our approach decides more than 91% of non-coverability instances, most of the time much faster when compared to existing tools, and none of those tools can individually decide more than 84%. Additionally, we show that our approach is also competitive when run on positive instances of coverability. In particular, overall our approach decides 142 out of 176 (80%) instances of our benchmark suite, while the best competitor only decides 122 (69%) instances.

1.2. Structure of this Paper

In Section 2, we define general notation, discrete and continuous Petri nets, and linear rational arithmetic, and we also recall some results from the literature. Section 3 is devoted to our logical characterization of the reachability relation for continuous Petri nets. In Section 4, we study the complexity of the inclusion, ε-liveness and home-state problems. Section 5 presents our algorithm for the coverability problem and its implementation. In Section 6, we conclude with a summary of our results and discuss some perspectives of our work.

2. PRELIMINARIES

2.1. General Notation

We denote by $\mathbb{Q}$, $\mathbb{Z}$ and $\mathbb{N}$ the set of *rationals*, *integers*, and *natural numbers*, respectively, and by $\mathbb{Q}_+$ the set of *non-negative rationals*. Throughout the whole paper, all numbers are encoded in binary, and rational numbers are encoded as pairs of integers. For $q = a/b \in \mathbb{Q}$, we denote by $\langle q \rangle \stackrel{\text{def}}{=} \lceil \log a \rceil + \lceil \log b \rceil + 1$ the number of *bits required to represent* q. Let $\mathbb{D} \subseteq \mathbb{Q}$ and I be a finite set of indices, we denote by $\mathbb{D}^I$ the *set of vectors indexed by* I. We write a vector $\boldsymbol{u}$ as $\boldsymbol{u} = (u_i)_{i \in I}$. Given vectors $\boldsymbol{u} = (u_i)_{i \in I}, \boldsymbol{v} = (v_i)_{i \in I} \in \mathbb{D}^I$, *addition* $\boldsymbol{u} + \boldsymbol{v}$ is defined component-wise, and this definition can be lifted to sets of vectors. Moreover, $\boldsymbol{u} \leq \boldsymbol{v}$ whenever $u_i \leq v_i$ for all $i \in I$, and $\boldsymbol{u} < \boldsymbol{v}$ whenever $\boldsymbol{u} \leq \boldsymbol{v}$ and $\boldsymbol{u} \neq \boldsymbol{v}$. The *support of* $\boldsymbol{v}$ is the set $[\![\boldsymbol{v}]\!] \stackrel{\text{def}}{=} \{i \in I : \boldsymbol{v}_i \neq 0\}$.

Given finite sets of indices I and J, and $\mathbb{D} \subseteq \mathbb{Q}$, $\mathbb{D}^{I \times J}$ denotes the set of *matrices* over $\mathbb{D}$ with rows and columns indexed by elements from I and J, respectively. Let

$\mathbf{M} \in \mathbb{D}^{I \times J}$, $I' \subseteq I$ and $J' \subseteq J$, we denote by $\mathbf{M}_{I' \times J'}$ the $\mathbb{D}^{I' \times J'}$ *sub-matrix* obtained from $\mathbf{M}$ whose row and columns indices are restricted to I' and J', respectively.

2.2. Petri Nets

In what follows, we introduce the syntax and semantics of Petri nets. While we provide a single syntax for nets, we introduce in this section a discrete semantics (i.e. in $\mathbb{N}$), and a continuous semantics (i.e. in $\mathbb{Q}_+$) in the next section.

Definition 2.1. A *Petri net* is a tuple $\mathcal{N} = (P, T, \mathbf{Pre}, \mathbf{Post})$, where P is a finite set of *places*; T is a finite set of *transitions* such that $P \cap T = \emptyset$; and $\mathbf{Pre}, \mathbf{Post} \in \mathbb{N}^{P \times T}$ are the *backward* and *forward incidence matrices*, respectively.

A (discrete) *marking* of $\mathcal{N}$ is a vector of $\mathbb{N}^P$. The *incidence matrix* $\mathbf{Incid}$ of $\mathcal{N}$ is the $P \times T$ integer matrix defined by

$$\mathbf{Incid} \stackrel{\text{def}}{=} \mathbf{Post} - \mathbf{Pre}.$$

For complexity purposes, we assume that a Petri net is encoded by enumerating the transitions of $\mathcal{N}$, where every transition of $\mathcal{N}$ is encoded as a tuple listing its non-zero entries in $\mathbf{Pre}$ and $\mathbf{Post}$. We denote by $|\mathcal{N}|$ the *size* of $\mathcal{N}$. The *reverse net* of $\mathcal{N}$ is $\mathcal{N}^{-1} \stackrel{\text{def}}{=} (P, T, \mathbf{Post}, \mathbf{Pre})$. Let $p \in P$ and $t \in T$, the *pre-sets* of p and t are the sets ${}^\bullet p \stackrel{\text{def}}{=} \{t' \in T : \mathbf{Post}(p, t') > 0\}$ and ${}^\bullet t \stackrel{\text{def}}{=} \{p' \in P : \mathbf{Pre}(p', t) > 0\}$, respectively. Likewise, the *post-sets* of p and t are $p^\bullet \stackrel{\text{def}}{=} \{t' \in T : \mathbf{Pre}(p, t') > 0\}$ and $t^\bullet = \{p' \in P : \mathbf{Post}(p', t) > 0\}$, respectively. Those definitions can canonically be lifted to subsets of places and of transitions, e.g., for $Q \subseteq P$ we have ${}^\bullet Q = \bigcup_{p \in Q} {}^\bullet p$. We also introduce the *neighbors* of a subset of places/transitions by: ${}^\bullet Q^\bullet = {}^\bullet Q \cup Q^\bullet$. Let $S \subseteq T$, then $\mathcal{N}_S$ is the *sub-net* defined by $\mathcal{N}_S \stackrel{\text{def}}{=} ({}^\bullet S^\bullet, S, \mathbf{Pre}_{{}^\bullet S^\bullet \times S}, \mathbf{Post}_{{}^\bullet S^\bullet \times S})$.

We say that a transition $t \in T$ is *enabled* at a marking $\boldsymbol{m}$ whenever $\boldsymbol{m}(p) \geq \mathbf{Pre}(p, t)$ for every $p \in {}^\bullet t$. A transition t that is enabled can be *fired*, leading to a new marking $\boldsymbol{m}'$ such that for all places $p \in P$, $\boldsymbol{m}'(p) = \boldsymbol{m}(p) + \mathbf{Incid}(p, t)$. We write $\boldsymbol{m} \xrightarrow{t} \boldsymbol{m}'$ whenever t is enabled at $\boldsymbol{m}$ and leads to $\boldsymbol{m}'$, and write $\boldsymbol{m} \to \boldsymbol{m}'$ if $\boldsymbol{m} \xrightarrow{t} \boldsymbol{m}'$ for some $t \in T$. By $\to^*$ we denote the reflexive transitive closure of $\to$. A word $\sigma = t_1 t_2 \cdots t_k \in T^*$ is a *firing sequence of* $(\mathcal{N}, \boldsymbol{m}_0)$ whenever there exist markings $\boldsymbol{m}_1, \ldots, \boldsymbol{m}_k$ such that

$$\boldsymbol{m}_0 \xrightarrow{t_1} \boldsymbol{m}_1 \xrightarrow{t_2} \cdots \xrightarrow{t_{k-1}} \boldsymbol{m}_{k-1} \xrightarrow{t_k} \boldsymbol{m}_k.$$

Two of the most prominent decision problems for Petri nets are reachability and coverability.

Definition 2.2. Given a Petri net $\mathcal{N} = (P, T, \mathbf{Pre}, \mathbf{Post})$, an initial marking $\boldsymbol{m}_0 \in \mathbb{N}^P$ and a target marking $\boldsymbol{m} \in \mathbb{N}^P$, *reachability* is the problem to decide whether $\boldsymbol{m}_0 \to^* \boldsymbol{m}$, and the *coverability* problem asks whether $\boldsymbol{m}_0 \to^* \boldsymbol{m}'$ for some $\boldsymbol{m}' \geq \boldsymbol{m}$.

Reachability is known to be decidable, EXPSPACE-hard [Lipton 1976; Cardoza et al. 1976] and in $\mathbf{F}_{\omega^3}$ [Leroux and Schmitz 2015], a non-primitive-recursive complexity class. Coverability is EXPSPACE-complete [Lipton 1976; Cardoza et al. 1976; Rackoff 1978]. The *reachability relation* of a Petri net $\mathcal{N}$ is defined as

$$\mathcal{R}(\mathcal{N}) \stackrel{\text{def}}{=} \{(\boldsymbol{m}, \boldsymbol{m}') \in \mathbb{N}^P \times \mathbb{N}^P : \boldsymbol{m} \to^* \boldsymbol{m}'\}.$$

2.3. Continuous Petri Nets

Continuous Petri nets are Petri nets in which markings may consist of rational numbers[1], and in which transitions may be fired a fractional number of times. Formally, a marking of a continuous Petri net is a vector $\boldsymbol{m} \in \mathbb{Q}_+^P$. Let $t \in T$, the *enabling degree* of t with respect to $\boldsymbol{m}$ is a function $enab(t, \boldsymbol{m}) \in \mathbb{Q}_+ \cup \{\infty\}$ defined by:

$$enab(t, \boldsymbol{m}) \stackrel{\text{def}}{=} \begin{cases} \min\{\boldsymbol{m}(p)/\mathbf{Pre}(p,t) : p \in {}^\bullet t\} & \text{if } {}^\bullet t \neq \emptyset \\ \infty & \text{otherwise.} \end{cases}$$

We say that t is $\mathbb{Q}$-*enabled* at $\boldsymbol{m}$ if $enab(t, \boldsymbol{m}) > 0$. If t is $\mathbb{Q}$-enabled it may be *fired* by any amount $q \in \mathbb{Q}_+$ such that $0 \leq q \leq enab(t, \boldsymbol{m})$, leading to a new marking $\boldsymbol{m}'$ such that for all places $p \in P$, $\boldsymbol{m}'(p) \stackrel{\text{def}}{=} \boldsymbol{m}(p) + q \cdot \mathbf{Incid}(p,t)$. In this case, we write $\boldsymbol{m} \xrightarrow{q \cdot t}_{\mathbb{Q}} \boldsymbol{m}'$. The definition of a $\mathbb{Q}$-*firing sequence* $\sigma = q_1 t_1 \cdots q_k t_k \in (\mathbb{Q}_+ \times T)^*$ is analogous to the standard definition of firing sequence, and so are $\to_{\mathbb{Q}}$, $\to_{\mathbb{Q}}^*$ and $\mathbb{Q}$-reachability. The $\mathbb{Q}$-*Parikh image* of the firing sequence σ is the vector $\pi(\sigma) \in \mathbb{Q}_+^T$ such that $\pi(\sigma)(t) \stackrel{\text{def}}{=} \sum_{t_i = t} q_i$. We also adapt the aforementioned decision problems for Petri nets.

Definition 2.3. Given a Petri net $\mathcal{N} = (P, T, \mathbf{Pre}, \mathbf{Post})$, an initial marking $\boldsymbol{m}_0 \in \mathbb{Q}_+^P$ and a target marking $\boldsymbol{m} \in \mathbb{Q}_+^P$, the $\mathbb{Q}$-*reachability* (respectively $\mathbb{Q}$-*coverability*) problem asks whether $\boldsymbol{m}_0 \to_{\mathbb{Q}}^* \boldsymbol{m}$ (respectively $\boldsymbol{m}_0 \to_{\mathbb{Q}}^* \boldsymbol{m}'$ for some $\boldsymbol{m}' \geq \boldsymbol{m}$).

Both reachability and coverability in continuous Petri nets can be decided with surprisingly low complexity, as captured in the following proposition.

PROPOSITION 2.4 ([FRACA AND HADDAD 2015]). *Both $\mathbb{Q}$-reachability and $\mathbb{Q}$-coverability are* P*-complete.*

The *continuous reachability relation* of a Petri net $\mathcal{N}$ is defined as

$$\mathbb{Q}\mathcal{R}(\mathcal{N}) \stackrel{\text{def}}{=} \{(\boldsymbol{m}, \boldsymbol{m}') \in \mathbb{Q}_+^P \times \mathbb{Q}_+^P : \boldsymbol{m} \to_{\mathbb{Q}}^* \boldsymbol{m}'\}.$$

An important observation about continuous Petri nets is that their reachability sets over approximate the reachability sets of discrete Petri nets: $\boldsymbol{m} \to \boldsymbol{m}'$ implies $\boldsymbol{m} \to_{\mathbb{Q}} \boldsymbol{m}'$, and hence $\boldsymbol{m} \to^* \boldsymbol{m}'$ implies $\boldsymbol{m} \to_{\mathbb{Q}}^* \boldsymbol{m}'$, i.e., $\mathcal{R}(\mathcal{N}) \subseteq \mathbb{Q}\mathcal{R}(\mathcal{N})$ for any Petri net $\mathcal{N}$.

2.4. Linear Rational Arithmetic

A tool allowing us to show decidability and complexity results in this paper is the first-order theory of the rational numbers with addition and order, $\mathrm{FO}\langle \mathbb{Q}, +, < \rangle$. *Atomic formulas* in this theory are linear constraints over first-order variables $\boldsymbol{x} = (x_1, \ldots, x_n)$ that we write as $\boldsymbol{a} \cdot \boldsymbol{x} \sim b$, where $\sim \in \{=, <, \leq, \geq, >\}$, $\boldsymbol{a} \in \mathbb{Q}^n$ and $b \in \mathbb{Q}$. The *size* $|\Phi|$ of a formula Φ is the number of symbols required to write down Φ, where we assume binary encoding of numbers. Formulas of $\mathrm{FO}\langle \mathbb{Q}, +, < \rangle$ are interpreted in their natural semantics, and we write $[\![\Phi(\boldsymbol{x})]\!]$ for the subset of n-tuples of rational numbers defined by Φ, i.e.,

$$[\![\Phi(\boldsymbol{x})]\!] \stackrel{\text{def}}{=} \{(q_1, \ldots, q_n) \in \mathbb{Q}^n : \Phi(q_1/x_1, \ldots, q_n/x_n) \text{ is valid}\}.$$

We write $\varphi(\boldsymbol{x}) \equiv \psi(\boldsymbol{x})$ whenever φ and ψ are *semantically equivalent*, i.e., define the same tuples of rational numbers. It is an easy exercise to show that for an $\mathrm{FO}\langle \mathbb{Q}, +, < \rangle$ formula, we may with no loss of generality assume that it does not contain any negation symbol, and that the only relation symbols used are $>$ and $\geq$.

[1] In fact, the original definition of David and Alla [1987] allows for real numbers, hence the name *continuous* Petri nets. However for studying decidability and complexity issues, rational numbers are more convenient.

The full theory of $\mathrm{FO}\langle\mathbb{Q},+,<\rangle$ is decidable in EXPSPACE and in fact complete for $\mathsf{STA}(*, 2^{O(n)}, n)$ [Berman 1980]. Here, $\mathsf{STA}(s(n), t(n), a(n))$ is the space-time-alternation measure on the complexity of a decision problem, which consists of all problems decidable by an alternating Turing machine that uses on every computation path at most $s(n)$ tape cells, runs in time $t(n)$ and makes $a(n)$ alternations. Moreover, "$*$" indicates an unbounded availability of a certain resource. For fixed quantifier alternation prefixes, the complexity of $\mathrm{FO}\langle\mathbb{Q},+,<\rangle$ is more manageable, and its restriction to a fixed number of i quantifier alternations is only complete for the i-th level of the polynomial hierarchy.

PROPOSITION 2.5 ([SONTAG 1985, COR. 3.4]). *For every $i > 0$, the Σ_i-fragment (resp. Π_i-fragment) of $\mathrm{FO}\langle\mathbb{Q},+,<\rangle$ is complete for Σ_i^{P} (resp. Π_i^{P}).*

3. A LOGICAL CHARACTERIZATION OF REACHABILITY IN CONTINUOUS PETRI NETS

In this section, we develop a logical characterization of the continuous reachability relation of a given Petri net in the existential fragment of $\mathrm{FO}\langle\mathbb{Q},+,<\rangle$. Our starting point is the work of Fraca and Haddad [2015] that presents an algorithm to decide reachability in continuous Petri nets in polynomial time. This algorithm builds upon a characterization of continuous reachability that we recall in Section 3.1 below. Subsequently, in Section 3.2 we show how this characterization can be used in order to obtain an existential $\mathrm{FO}\langle\mathbb{Q},+,<\rangle$ formula of linear size defining the continuous reachability relation. For the remainder of this section, we fix a Petri net $\mathcal{N} = (P, T, \mathbf{Pre}, \mathbf{Post})$.

3.1. Three Criteria Characterizing Continuous Reachability

The key insight underlying the algorithm developed by Fraca and Haddad is that continuous reachability can be characterized in terms of three simple criteria. First, for a Petri net $\mathcal{N}$ and a marking $\boldsymbol{m} \in \mathbb{Q}_+^P$, we introduce the auxiliary definition of *firing set*, denoted by $fs(\mathcal{N}, \boldsymbol{m})$. Recall that $\pi(\sigma)$ denotes the Parikh image of a firing sequence σ, and that $[\![\pi(\sigma)]\!] \subseteq T$ is the support of this firing sequence. We set

$$fs(\mathcal{N}, \boldsymbol{m}) \stackrel{\text{def}}{=} \{[\![\pi(\sigma)]\!] : \sigma \in (\mathbb{Q}_+ \times T)^* \text{ and there is } \boldsymbol{m}' \in \mathbb{Q}_+^P \text{ s.t. } \boldsymbol{m} \xrightarrow{\sigma}_{\mathbb{Q}} \boldsymbol{m}'\}.$$

Thus, $fs(\mathcal{N}, \boldsymbol{m})$ is the set of supports of firing sequences starting in an initial marking $\boldsymbol{m}$. Even though $fs(\mathcal{N}, \boldsymbol{m})$ can be of size exponential with respect to $|T|$, deciding $T' \in fs(\mathcal{N}, \boldsymbol{m})$ for some $T' \subseteq T$ can be done in polynomial time. The following proposition characterizes the pairs of $\mathbb{Q}$-reachable markings.

PROPOSITION 3.1 ([FRACA AND HADDAD 2015, THM. 20]). *Given a Petri net $\mathcal{N} = (P, T, \mathbf{Pre}, \mathbf{Post})$, we have $(\boldsymbol{m}, \boldsymbol{m}') \in \mathbb{Q}\mathcal{R}(\mathcal{N})$ if, and only if, there exists $\boldsymbol{y} \in \mathbb{Q}_+^T$ such that*

(i) $\boldsymbol{m}' = \boldsymbol{m} + \mathbf{Incid} \cdot \boldsymbol{y}$
(ii) $[\![\boldsymbol{y}]\!] \in fs(\mathcal{N}, \boldsymbol{m})$
(iii) $[\![\boldsymbol{y}]\!] \in fs(\mathcal{N}^{-1}, \boldsymbol{m}')$.

Here, $\boldsymbol{y}$ should be seen as the Parikh image of a firing sequence. The first item expresses the *state equation* of $\mathcal{N}$ with respect to $\boldsymbol{m}$, $\boldsymbol{m}'$ and $\boldsymbol{y}$; the two subsequent items express that the support of the solution of the state equation has to lie in the firing set of $\mathcal{N}$ and its reverse.

3.2. The Logical Characterization

We now show how to encode the conditions of Proposition 3.1 in the existential fragment of $\mathrm{FO}\langle\mathbb{Q},+,<\rangle$. Condition (i) of Proposition 3.1, which expresses the state equation, is readily expressed as a system of linear equations and thus directly corresponds to a formula $\Phi_{eqn}^{\mathcal{N}}(\boldsymbol{x}, \boldsymbol{x}', \boldsymbol{y})$ which holds whenever a marking $\boldsymbol{x}'$ is reached starting in

marking $\boldsymbol{x}$ by firing every transition $\boldsymbol{y}(t)$ times (without any consideration whether such a firing sequence would actually be admissible). Formally,

$$\Phi^{\mathcal{N}}_{eqn}(\boldsymbol{x}, \boldsymbol{x}', \boldsymbol{y}) \stackrel{\text{def}}{=} \boldsymbol{x}' = \boldsymbol{x} + \mathbf{Incid} \cdot \boldsymbol{y} \wedge \boldsymbol{y} \geq \mathbf{0},$$

where $\mathbf{0}$ denotes the null-vector in an appropriate dimension.

Next, we show how to encode Conditions (ii) and (iii) into suitable formulas. To this end, we require an effective characterization of membership in the firing set $fs(\mathcal{N}, \boldsymbol{x})$ defined in Section 3.1. The following characterization can be derived from [Fraca and Haddad 2015, Cor. 19]. First, we define a monotonic increasing function $incfs_{\mathcal{N},\boldsymbol{x}} : 2^T \to 2^T$ as follows:

$$incfs_{\mathcal{N},\boldsymbol{x}}(S) \stackrel{\text{def}}{=} S \cup \{t \in T : {}^\bullet t \subseteq (\llbracket \boldsymbol{x} \rrbracket \cup \{s^\bullet : s \in S\})\}\,.$$

Loosely speaking, $incfs_{\mathcal{N},\boldsymbol{x}}(S)$ returns the set of transitions S, and additionally those transitions that can be fired when the places in $\llbracket \boldsymbol{x} \rrbracket$ and those that receive tokens from transitions in S all carry tokens. Fraca and Haddad [2015, Cor. 19] considered an algorithm from which it follows that

$$T' \in fs(\mathcal{N}, \boldsymbol{x}) \iff T' = \mathrm{lfp}(incfs_{\mathcal{N}_{T'},\boldsymbol{x}}), \tag{1}$$

where lfp is the least fixed point operator, i.e.,

$$T' = incfs_{\mathcal{N}_{T'},\boldsymbol{x}}(\cdots(incfs_{\mathcal{N}_{T'},\boldsymbol{x}}(\emptyset))\cdots).$$

Notice the restriction of $\mathcal{N}$ to $\mathcal{N}_{T'}$. Clearly, the least fixed point is reached after at most $|T'|$ iterations.

In order to decide whether $\llbracket \boldsymbol{y} \rrbracket \in fs(\mathcal{N}, \boldsymbol{x})$, we simulate this fixed-point computation by an existential FO$\langle \mathbb{Q}, +, > \rangle$-formula $\Phi^{\mathcal{N}}_{fs}(\boldsymbol{x}, \boldsymbol{y})$. Our approach is inspired by a technique of Verma, Seidl and Schwentick that was used to show that the reachability relation for communication-free Petri nets is definable by a formula in existential Presburger arithmetic of linear size [Verma et al. 2005]. The basic idea is to introduce additional first-order variables $\boldsymbol{z}$ indexed by $P \cup T$ that, given a firing set, capture the relative order in which transitions of this set are fired and the order in which their input places are marked. This order corresponds to the computation of $\mathrm{lfp}(incfs_{\mathcal{N}_{\llbracket \boldsymbol{y} \rrbracket},\boldsymbol{x}})$ and is encoded via a numerical value $\boldsymbol{z}(t)$ (respectively $\boldsymbol{z}(p)$), representing an index that must be strictly greater than zero for a transition (respectively an input place of a transition) of this set. In addition, input places have to be marked before the firing of a transition, which is captured by the following formula:

$$\Phi^{\mathcal{N}}_{dt}(\boldsymbol{y}, \boldsymbol{z}) \stackrel{\text{def}}{=} \bigwedge_{t \in T} \left(\boldsymbol{y}(t) > 0 \to \left(\boldsymbol{z}(t) > 0 \wedge \bigwedge_{p \in {}^\bullet t} 0 < \boldsymbol{z}(p) \leq \boldsymbol{z}(t) \right) \right).$$

Moreover, a place is either marked initially or after the firing of a transition of the firing set. Hence

$$\Phi^{\mathcal{N}}_{mk}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \stackrel{\text{def}}{=} \bigwedge_{p \in P} \left(\boldsymbol{z}(p) > 0 \to \left(\boldsymbol{x}(p) > 0 \vee \bigvee_{t \in {}^\bullet p} \boldsymbol{y}(t) > 0 \wedge \boldsymbol{z}(t) < \boldsymbol{z}(p) \right) \right).$$

We can now take the conjunction of the formulas above in order to obtain a logical characterization of $fs(\mathcal{N}, \boldsymbol{w})$:

$$\Phi^{\mathcal{N}}_{fs}(\boldsymbol{x}, \boldsymbol{y}) \stackrel{\text{def}}{=} \exists \boldsymbol{z} : \Phi^{\mathcal{N}}_{dt}(\boldsymbol{y}, \boldsymbol{z}) \wedge \Phi^{\mathcal{N}}_{mk}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}).$$

Having logically characterized all conditions of Proposition 3.1, we can define the global continuous reachability relation for a Petri net $\mathcal{N}$ as follows:

$$\Phi_{\mathcal{N}}(\boldsymbol{x},\boldsymbol{x}') \stackrel{\text{def}}{=} \exists \boldsymbol{y} : \Phi^{\mathcal{N}}_{eqn}(\boldsymbol{x},\boldsymbol{x}',\boldsymbol{y}) \wedge \Phi^{\mathcal{N}}_{fs}(\boldsymbol{x},\boldsymbol{y}) \wedge \Phi^{\mathcal{N}^{-1}}_{fs}(\boldsymbol{x}',\boldsymbol{y}).$$

PROPOSITION 3.2. *Let $\mathcal{N}$ be a Petri net. There exists an existential FO$\langle\mathbb{Q},+,<\rangle$-formula $\Phi_{\mathcal{N}}(\boldsymbol{x},\boldsymbol{x}')$ computable in time $O(|\mathcal{N}|)$ with $4\cdot|P|+3\cdot|T|$ variables such that $\mathbb{Q}\mathcal{R}(\mathcal{N}) = [\![\Phi_{\mathcal{N}}(\boldsymbol{x},\boldsymbol{x}')]\!]$.*

PROOF. We first show that $\mathbb{Q}\mathcal{R}(\mathcal{N}) \subseteq [\![\Phi_{\mathcal{N}}(\boldsymbol{x},\boldsymbol{x}')]\!]$. Suppose $(\boldsymbol{m},\boldsymbol{m}') \in \mathbb{Q}\mathcal{R}(\mathcal{N})$. By Proposition 3.1, there is some $\boldsymbol{y} \in \mathbb{Q}_+^T$ such that $\boldsymbol{m}' = \boldsymbol{m} + \mathbf{Incid}\cdot\boldsymbol{y}$, $[\![\boldsymbol{y}]\!] \in fs(\mathcal{N},\boldsymbol{m})$ and $[\![\boldsymbol{y}]\!] \in fs(\mathcal{N},\boldsymbol{m}')$. Consequently, $\Phi^{\mathcal{N}}_{eqn}(\boldsymbol{m}/\boldsymbol{x},\boldsymbol{m}'/\boldsymbol{x}',\boldsymbol{y})$ holds. Let $T' = [\![\boldsymbol{y}]\!]$, since $T' \in fs(\mathcal{N},\boldsymbol{m})$, by (1) we have $T' = \mathrm{lfp}(incfs_{\mathcal{N}_{T'},\boldsymbol{m}})$. Denote by $incfs^i_{\mathcal{N}_{T'},\boldsymbol{m}}(\emptyset)$ the i-fold application of $incfs_{\mathcal{N}_{T'},\boldsymbol{m}}$ on $\emptyset$. For every transition $t \in T$, we record the first time t occurs in the fixed point computation and define

$$\boldsymbol{z}(t) \stackrel{\text{def}}{=} \begin{cases} i & \text{if } t \in incfs^i_{\mathcal{N}_{T'},\boldsymbol{m}}(\emptyset) \text{ and } t \notin incfs^{i-1}_{\mathcal{N}_{T'},\boldsymbol{m}}(\emptyset) \\ 0 & \text{otherwise.} \end{cases}$$

Since $\mathrm{lfp}(incfs_{\mathcal{N}_{T'},\boldsymbol{m}}) = incfs^{|T'|}_{\mathcal{N}_{T'},\boldsymbol{m}}(\emptyset)$, $\boldsymbol{z}(t) > 0$ for all $t \in T'$, and $\boldsymbol{z}(t) = 0$ for every $t \in T \setminus T'$. For every $p \in P$, define

$$\boldsymbol{z}(p) \stackrel{\text{def}}{=} \begin{cases} i & \text{if } p \in {}^\bullet(incfs^i_{\mathcal{N}_{T'},\boldsymbol{m}}(\emptyset)) \text{ and } p \notin {}^\bullet(incfs^{i-1}_{\mathcal{N}_{T'},\boldsymbol{m}}(\emptyset)) \\ 0 & \text{otherwise.} \end{cases}$$

We now claim that $\boldsymbol{z}$ is a valid solution for $\Phi^{\mathcal{N}}_{dt}(\boldsymbol{y},\boldsymbol{z})$ and $\Phi^{\mathcal{N}}_{mk}(\boldsymbol{m}/\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$. Regarding the former, since $\boldsymbol{y}(t) > 0$ whenever $t \in T'$, by construction $\boldsymbol{z}(p) \leq \boldsymbol{z}(t)$ for every $p \in {}^\bullet t$, since $t \in incfs^{\boldsymbol{z}(t)}_{\mathcal{N}_{T'},\boldsymbol{m}}(\emptyset)$, and hence $p \in {}^\bullet(incfs^{\boldsymbol{z}(t)}_{\mathcal{N}_{T'},\boldsymbol{m}}(\emptyset))$. Likewise, it follows that $\boldsymbol{z}(p) > 0$, and, as already discussed above, $\boldsymbol{z}(t) > 0$. Regarding validity of $\Phi^{\mathcal{N}}_{mk}(\boldsymbol{m}/\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$, suppose $\boldsymbol{z}(p) > 0$. If $\boldsymbol{z}(p) = 1$ then by definition of $incfs_{\mathcal{N}_{T'},\boldsymbol{m}}$ we have $\boldsymbol{m}(p) > 0$. Otherwise, again by definition of $incfs_{\mathcal{N}_{T'},\boldsymbol{m}}$, there is some $t \in incfs^{\boldsymbol{z}(p)-1}_{\mathcal{N}_{T'},\boldsymbol{m}}(\emptyset)$ such that $p \in t^\bullet$, and hence $0 < \boldsymbol{z}(t) < \boldsymbol{z}(p)$. It follows that $\Phi^{\mathcal{N}}_{fs}(\boldsymbol{m}/\boldsymbol{x},\boldsymbol{y})$ is valid. Validity of $\Phi^{\mathcal{N}^{-1}}_{fs}(\boldsymbol{m}/\boldsymbol{x},\boldsymbol{y})$ can be shown along similar lines. It follows that $\Phi_{\mathcal{N}}(\boldsymbol{m}/\boldsymbol{x},\boldsymbol{m}'/\boldsymbol{x}')$ is valid.

In order to show $[\![\Phi_{\mathcal{N}}(\boldsymbol{x},\boldsymbol{x}')]\!] \subseteq \mathbb{Q}\mathcal{R}(\mathcal{N})$, let $\boldsymbol{x},\boldsymbol{x}',\boldsymbol{y}$, $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ be valuations of variables such that $\Phi_{\mathcal{N}}(\boldsymbol{x},\boldsymbol{x}')$ evaluates to true (here, we implicitly view $\Phi_{\mathcal{N}}$ to be open in all those variables). Again, Condition (i) of Proposition 3.1 is easily seen to be true due to $\Phi^{\mathcal{N}}_{eqn}(\boldsymbol{x},\boldsymbol{x}',\boldsymbol{y})$ being valid. Thus, let us show that Condition (ii) of Proposition 3.1 holds, i.e., that $[\![\boldsymbol{y}]\!] \in fs(\mathcal{N},\boldsymbol{x})$. Our starting point is that $\Phi^{\mathcal{N}}_{dt}(\boldsymbol{y},\boldsymbol{z}_1/\boldsymbol{z})$ and $\Phi_{mk}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}_1/\boldsymbol{z})$ hold. Let $T' \stackrel{\text{def}}{=} [\![\boldsymbol{y}]\!]$ and $T_1,\ldots,T_m \subseteq T'$ be such that

— $\boldsymbol{z}_1(t) = \boldsymbol{z}_1(t')$ whenever $\{t,t'\} \subseteq T_i$;
— $\boldsymbol{z}_1(t_i) < \boldsymbol{z}_1(t_j)$ for all $i < j$ such that $t_i \in T_i$ and $t_j \in T_j$; and
— $\boldsymbol{z}_1(t) > 0$ if and only if $t \in T_i$ for some $1 \leq i \leq m$.

We now show by induction on $i \geq 1$ that $\bigcup_{1\leq j\leq i} T_j \subseteq incfs^i_{\mathcal{N}_{T'},\boldsymbol{m}}(\emptyset)$.

For the induction base case, since $\Phi^{\mathcal{N}}_{dt}(\boldsymbol{y},\boldsymbol{z}_1/\boldsymbol{z})$ holds, for every transition $t \in T_1$ we have either ${}^\bullet t = \emptyset$, or $0 < \boldsymbol{z}_1(p) \leq \boldsymbol{z}_1(t)$ for all $p \in {}^\bullet t$. In the latter case, since $\Phi_{mk}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}_1/\boldsymbol{z})$ holds, we have $\boldsymbol{x}(p) > 0$, i.e., $p \in [\![\boldsymbol{x}]\!]$, for all $p \in {}^\bullet t$ since there cannot

Table I. Decision problems for continuous Petri nets $\mathcal{N}, \mathcal{M}$, where $\mathcal{N} = (P, T, \mathbf{Pre}, \mathbf{Post})$. Here, ε is a first-order variable and $\boldsymbol{x}, \boldsymbol{x}'$ and $\boldsymbol{x}''$ are vectors of first-order variables indexed by places $p \in P$, e.g., $\boldsymbol{x} = (x_p)_{p \in P}$. New complexity results established in this paper are colored.

Decision Problem	Definition	Complexity
Coverability	$\Phi^{\mathcal{N}}_{cvr}(\boldsymbol{x}, \boldsymbol{x}') \stackrel{\text{def}}{=} \exists \boldsymbol{x}'' : \Phi_{\mathcal{N}}(\boldsymbol{x}, \boldsymbol{x}'') \wedge \boldsymbol{x}'' \geq \boldsymbol{x}'$	P-complete
Boundedness	$\Phi_{bnd}(\boldsymbol{x}) \stackrel{\text{def}}{=} \exists \boldsymbol{x}' : \forall \boldsymbol{x}'' : \Phi_{\mathcal{N}}(\boldsymbol{x}, \boldsymbol{x}'') \rightarrow \boldsymbol{x}' \geq \boldsymbol{x}''$	P-complete
Inclusion	$\Phi^{\mathcal{N},\mathcal{M}}_{inc}(\boldsymbol{x}, \boldsymbol{x}') \stackrel{\text{def}}{=} \forall \boldsymbol{x}'' : \Phi_{\mathcal{N}}(\boldsymbol{x}, \boldsymbol{x}'') \rightarrow \Phi_{\mathcal{M}}(\boldsymbol{x}', \boldsymbol{x}'')$	coNP-complete
ε-Liveness	$\Phi^{\mathcal{N}}_{lv}(\boldsymbol{x}) \stackrel{\text{def}}{=} \exists \varepsilon : \varepsilon > 0 \wedge \forall \boldsymbol{x}' : \Phi_{\mathcal{N}}(\boldsymbol{x}, \boldsymbol{x}') \rightarrow \rightarrow \bigwedge_{t \in T} \left(\exists \boldsymbol{x}'' : \Phi_{\mathcal{N}}(\boldsymbol{x}', \boldsymbol{x}'') \wedge \Phi^{\mathcal{N},t}_{nbld}(\varepsilon, \boldsymbol{x}'') \right)$	in Σ^{P}_3
Struct. ε-Liveness	$\Phi^{\mathcal{N}}_{slv} \stackrel{\text{def}}{=} \exists \boldsymbol{x} : \Phi^{\mathcal{N}}_{lv}(\boldsymbol{x})$	in Σ^{P}_3
Home State	$\Phi^{\mathcal{N}}_{hm}(\boldsymbol{x}, \boldsymbol{x}') \stackrel{\text{def}}{=} \forall \boldsymbol{x}'' : \Phi_{\mathcal{N}}(\boldsymbol{x}, \boldsymbol{x}'') \rightarrow \Phi_{\mathcal{N}}(\boldsymbol{x}'', \boldsymbol{x}')$	coNP-complete
Exist. Home State	$\Phi^{\mathcal{N}}_{ehm}(\boldsymbol{x}) \stackrel{\text{def}}{=} \exists \boldsymbol{x}' : \Phi_{hm}(\boldsymbol{x}, \boldsymbol{x}')$	in Σ^{P}_2

be transitions $t' \in T'$ such that $0 < \boldsymbol{z}_1(t') < \boldsymbol{z}_1(p)$ due to all transitions in T_1 being minimal with respect to $\boldsymbol{z}_1$. Hence $T_1 \subseteq incfs^1_{\mathcal{N}_{T'}, \boldsymbol{m}}(\emptyset)$.

For the induction step, suppose that $\bigcup_{1 \leq j \leq i} T_j \subseteq incfs^i_{\mathcal{N}_{T'}, \boldsymbol{m}}(\emptyset)$ and let $t \in T_{i+1}$. As in the base case, if ${}^\bullet t = \emptyset$ we are done. Otherwise, since $\Phi^{\mathcal{N}}_{dt}(\boldsymbol{y}, \boldsymbol{z}_1/\boldsymbol{z})$ holds, we have $0 < \boldsymbol{z}_1(p) \leq \boldsymbol{z}_1(t)$ for all $p \in {}^\bullet t$. Since $\Phi_{mk}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}_1/\boldsymbol{z})$ also holds, for $p \in {}^\bullet t$ we have $\boldsymbol{x}(p) > 0$, or there is some $t' \in \bigcup_{1 \leq j \leq i} T_j$ such that $p \in t'^\bullet$. By the induction hypothesis, we conclude that ${}^\bullet t \subseteq (\llbracket \boldsymbol{x} \rrbracket \cup \{ t'^\bullet : t' \in incfs^i_{\mathcal{N}_{T'}, \boldsymbol{m}}(\emptyset) \})$, and hence $t \in incfs^{i+1}_{\mathcal{N}_{T'}, \boldsymbol{m}}(\emptyset)$.

Now since $\boldsymbol{y}(t) > 0$ implies $\boldsymbol{z}(t) > 0$, we have $T' = \bigcup_{1 \leq i \leq m} T_i$, and consequently $T' = \llbracket \boldsymbol{y} \rrbracket \in fs(\mathcal{N}, \boldsymbol{x})$.

Regarding the complexity of computing $\Phi_{\mathcal{N}}(\boldsymbol{x}, \boldsymbol{x}')$, we first note that in linear time we can compute a list that keeps for each place $p \in P$ its set of incoming transitions, ${}^\bullet p$, and its set of outgoing transitions, $p^\bullet$. It follows that every formula $\Phi^{\mathcal{N}}_{eqn}(\boldsymbol{x}, \boldsymbol{x}', \boldsymbol{y})$, $\Phi^{\mathcal{N}}_{fs}(\boldsymbol{x}, \boldsymbol{y})$ and $\Phi^{\mathcal{N}^{-1}}_{fs}(\boldsymbol{x}', \boldsymbol{y})$ can be computed by traversing the encoding of $\mathcal{N}$ or the aforementioned generated list once, and hence $\Phi_{\mathcal{N}}(\boldsymbol{x}, \boldsymbol{x}')$ can overall be computed in $O(|\mathcal{N}|)$. Finally, observe that both $\boldsymbol{x}$ and $\boldsymbol{x}'$ are vectors of $|P|$ variables each, $\boldsymbol{y}$ is a vector of $|T|$ variables, and $\boldsymbol{z}$ is a vector of $|P| + |T|$ variables occurring twice. Consequently, $\Phi_{\mathcal{N}}(\boldsymbol{x}, \boldsymbol{x}')$ contains $4 \cdot |P| + 3 \cdot |T|$ variables. □

4. DECISION PROBLEMS FOR CONTINUOUS PETRI NETS

Besides reachability, there exists a plethora of further decision problems for Petri nets whose decidability and computational complexity has been studied in the literature, both in the continuous and discrete setting. In Table I, we employ the formula $\Phi_{\mathcal{N}}(\boldsymbol{x}, \boldsymbol{x}')$ defining the continuous reachability relation from Proposition 3.2 in order to give an overview of some decision problems for continuous Petri nets, their formal definition in FO$\langle \mathbb{Q}, +, < \rangle$, and their computational complexity. Cells with blue background color indicate results that we will establish in the subsequent sections.

In words, given a Petri net $\mathcal{N}$, an initial marking $\boldsymbol{m}$ and a marking $\boldsymbol{m}'$, coverability asks whether there is some configuration $\boldsymbol{m}''$ that is reachable and in which every place contains at least as many tokens as specified by $\boldsymbol{m}'$. This problem is P-complete [Fraca and Haddad 2015]. *Boundedness* asks whether there is some marking $\boldsymbol{m}'$ such that every marking $\boldsymbol{m}''$ reachable from $\boldsymbol{m}$ does not exceed $\boldsymbol{m}'$ in any of its components. This problem is also P-complete [Fraca and Haddad 2015]. Given another

Petri net $\mathcal{M}$ with the same set of places as $\mathcal{N}$, *inclusion* asks whether every marking $\boldsymbol{m}''$ reachable from $\boldsymbol{m}$ in $\mathcal{N}$ is also reachable in $\mathcal{M}$ starting in $\boldsymbol{m}'$. This problem is known to be coNP-hard and in EXP [Fraca and Haddad 2015], and we improve this result in Section 4.2 and show that inclusion is actually coNP-complete. *ε-Liveness* asks whether there exists a rational $\varepsilon > 0$ such that for every marking $\boldsymbol{m}'$ reachable from $\boldsymbol{m}$ and every transition $t \in T$, a marking $\boldsymbol{m}''$ is reachable such that t has enabling degree at least ε in $\boldsymbol{m}''$. For $\boldsymbol{x} = (x_p)_{p \in P}$, Table I uses the abbreviation

$$\Phi^{\mathcal{N},t}_{nbld}(\varepsilon, \boldsymbol{x}) \stackrel{\text{def}}{=} \begin{cases} \bigwedge_{p \in {}^\bullet t} \frac{1}{\mathbf{Pre}(p,t)} \cdot x_p \geq \varepsilon & \text{if } {}^\bullet t \neq \emptyset \\ \mathit{true} & \text{otherwise,} \end{cases}$$

i.e., $\Phi^{\mathcal{N},t}_{nbld}(\varepsilon, \boldsymbol{x})$ holds whenever the enabling degree of t in $\boldsymbol{x}$ is at least ε. *Structural ε-liveness* asks whether there exists some initial marking $\boldsymbol{m}$ such that $\mathcal{N}$ is ε-live in $\boldsymbol{m}$. We show in Section 4.3 that ε-liveness and structural ε-liveness are decidable in Σ^{P}_3. As mentioned by Recalde et al. [1999, Sec. 5], the definition of (structural) ε-liveness is arguably more suitable for continuous Petri nets as compared to the standard definitions from discrete Petri nets which only require enabledness. For example, consider a Petri net such that all of its transitions decrease the number of tokens. Such a Petri net is not live under the discrete semantics. However, it could be "live" under the continuous semantics by firing transitions by increasingly smaller amounts. It is difficult to give a natural interepration justifying such Zeno runs to contribute to liveness, especially in such a system that strictly monotonically decreases the number of tokens of every place. Finally, a marking $\boldsymbol{m}'$ is a *home state* if $\boldsymbol{m}'$ can be reached from every marking $\boldsymbol{m}''$ that can be reached from $\boldsymbol{m}$. The *existential home-state problem* is to decide whether there exists a home state for a given initial marking $\boldsymbol{m}$. In Section 4.4, we show that the home-state problem is coNP-complete, and that the existential home-state problem is decidable in Σ^{P}_2.

The benefit of our characterization of continuous reachability in $\mathrm{FO}\langle \mathbb{Q}, +, < \rangle$ is that we immediately obtain decidability of all decision problems presented in Table I via Proposition 2.5, albeit not always with optimal upper bounds. Roughly speaking, the reason for this is that satisfiability in existential $\mathrm{FO}\langle \mathbb{Q}, +, < \rangle$ is NP-complete while reachability in continuous Petri nets is P-complete. Nevertheless, our characterization turns out to be a good starting point for the (tight) upper bounds that we develop.

In the discrete setting, the complexity of the decision problems we consider is only rarely known and significantly higher than in the continuous setting. Both coverability and boundedness are EXPSPACE-complete [Lipton 1976; Rackoff 1978]; inclusion is undecidable [Hack 1976]; standard liveness is inter-reducible with reachability [Hack 1974] and structural liveness has recently been shown decidable and hard for reachability [Jančar 2017]; the home state problem is decidable [de Frutos Escrig and Johnen 1989; Desel and Esparza 1995], and the existential home-state problem has recently been shown decidable and hard for reachability in discrete Petri nets [Best and Esparza 2016].

It is worth mentioning that, except for coverability, the discrete versions of the decision problems of Table I cannot be over approximated via the continuous semantics. Let $\mathcal{N}_1, \mathcal{N}_2$ and $\mathcal{N}_3$ be the Petri nets illustrated in Figure 1 from left to right. It is readily seen that

$$\mathcal{R}(\mathcal{N}_1)(1,0) = \{(1,0)\}, \qquad \mathcal{R}(\mathcal{N}_2)(1,0) = \{(n+1,0) : n \in \mathbb{N}\},$$
$$\mathbb{Q}\mathcal{R}(\mathcal{N}_1)(1,0) = \{(1,n) : n \in \mathbb{Q}_+\}, \qquad \mathbb{Q}\mathcal{R}(\mathcal{N}_2)(1,0) = \{(n+1,0) : n \in \mathbb{Q}_+\}.$$

We have $\mathcal{R}(\mathcal{N}_1)(1,0) \subseteq \mathcal{R}(\mathcal{N}_2)(1,0)$, but $\mathbb{Q}\mathcal{R}(\mathcal{N}_1)(1,0) \not\subseteq \mathbb{Q}\mathcal{R}(\mathcal{N}_2)(1,0)$. Moreover $\mathcal{R}(\mathcal{N}_1)(1,0)$ is bounded, but $\mathbb{Q}\mathcal{R}(\mathcal{N}_1)(1,0)$ is not. Clearly, $(1,0)$ is a discrete home state

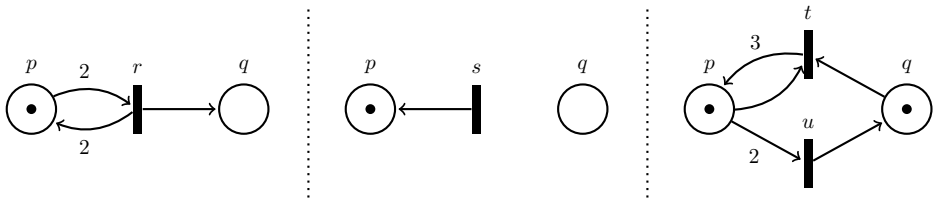

Fig. 1. Examples of Petri nets $\mathcal{N}_1$, $\mathcal{N}_2$ and $\mathcal{N}_3$ showing that boundedness, inclusion, liveness and home states problems cannot be over approximated by the continuous semantics. The rightmost Petri net is borrowed from Recalde et al. [1999].

of $\mathcal{N}_1$ from $(1,0)$ since it is the unique reachable marking. However, $\mathcal{N}_1$ has no continuous home state from $(1,0)$ since r strictly increases the number of tokens. Therefore, inclusion, boundedness and the (existential) home state problems cannot be over approximated by the continuous semantics. As for liveness, observe that $\mathcal{N}_3$ is live from $(1,1)$ as its only discrete run is $(1,1) \xrightarrow{t} (3,0) \xrightarrow{u} (1,1) \xrightarrow{t} \cdots$. However, for every marking (m,n), we have $(m,n) \xrightarrow{(m/2)u}_{\mathbb{Q}} (0, m/2+n)$ and $(0, m/2+n)$ is dead, regardless of the enabling degree.

4.1. Small Points in Convex Polyhedra and Projections of Convex Polyhedra

As a preparatory step for the subsequent sections, here we recall and establish a number of results concerning linear programming and $\mathrm{FO}\langle\mathbb{Q},+,<\rangle$. We first show that nonempty systems of mixed linear inequalities and strict inequalities contain small points. Here and in the following, by "small" we mean objects which can be represented using a polynomial number of bits in the representation of a given system. Next, we show that the set of solutions of such systems can be obtained as the union of solutions sets of a finite number of polyhedra of small facet complexity. Finally, we provide a quantifier elimination method for $\mathrm{FO}\langle\mathbb{Q},+,<\rangle$ which yields formulas of small facet complexity. All terminology will be clarified below.

The results explored in this section have partly been obtained by Sontag [1985] and Schrijver [1998], and are not novel as such. However, when looking at the exposition of Sontag [1985], we observed that some proof details are rather coarse, and that more detailed yet no more complex arguments are possible. In Proposition 4.4 below, compared to the work of Sontag [1985], we also provide a more explicit geometric characterization of projections of systems of mixed linear inequalities and strict inequalities. We believe and hope that an interested reader will view and appreciate our exposition as a valuable complement to the results and reasoning presented by Sontag [1985].

We begin with introducing some auxiliary definitions. Let $A \in \mathbb{Q}^{m\times n}$ be an $(m \times n)$-matrix with rational coefficients, and let $\boldsymbol{c} \in \mathbb{Q}^m$. We call $S : A \cdot \boldsymbol{x} \geq \boldsymbol{c}$ a system of linear inequalities in the unknowns $\boldsymbol{x} = (x_i)_{1\leq i\leq n}$. Writing $A = (a_{i,j})_{1\leq i\leq m, 1\leq j\leq n}$ and $\boldsymbol{c} = (c_i)_{1\leq i\leq m}$, S can alternatively be viewed as the conjunction

$$\Psi_S(\boldsymbol{x}) \stackrel{\text{def}}{=} \bigwedge_{1\leq i\leq m} (a_{i,1}, \ldots, a_{i,n}) \cdot \boldsymbol{x} \geq c_i. \tag{2}$$

We define the set $[\![S]\!]$ of solutions of S as $[\![S]\!] \stackrel{\text{def}}{=} [\![\Psi_S(\boldsymbol{x})]\!] \subseteq \mathbb{Q}^n$. The facet complexity $\langle S\rangle$ of S is the largest bit size needed to represent any row of S; formally:

$$\langle S\rangle \stackrel{\text{def}}{=} 1 + \max_{1\leq i\leq m} \sum_{1\leq j\leq n} \langle a_{i,j}\rangle + \langle c_i\rangle.$$

Note that the facet complexity is independent of the number of rows of S. For a single vector $\boldsymbol{v} \in \mathbb{Q}^m$, $\langle \boldsymbol{v} \rangle$ is the bit size of the largest component of $\boldsymbol{v}$. By $\|\boldsymbol{v}\|$ we denote the maximum of the absolute values of all components of $\boldsymbol{v}$, and for finite sets $V \subseteq \mathbb{Q}^m$ we define $\|V\| \stackrel{\text{def}}{=} \max_{\boldsymbol{v} \in V} \|\boldsymbol{v}\|$.

If in the representation in (2) we allow further relational symbols such as $>$ and $=$, we obtain systems of mixed linear inequalities, strict inequalities and equalities. We write such systems, e.g., as $S : A \cdot \boldsymbol{x} \geq \boldsymbol{c} \wedge B \cdot \boldsymbol{x} > \boldsymbol{d}$ (in particular, an equality can always be written as a conjunction of two non-strict inequalities). The definition of $\langle S \rangle$ is analogous as above.

Given a finite set of vectors $V = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\} \subseteq \mathbb{Q}^m$, in this paper the convex hull and the cone generated by V are defined as

$$\operatorname{conv} V \stackrel{\text{def}}{=} \left\{ \boldsymbol{v} \in \mathbb{Q}^m : \boldsymbol{v} = \sum_{i=1}^{n} \lambda_i \cdot \boldsymbol{v}_i, \lambda_i \in \mathbb{Q}, 0 \leq \lambda_i \leq 1, \sum_{i=1}^{n} \lambda_i \leq 1 \right\}$$

$$\operatorname{cone} V \stackrel{\text{def}}{=} \left\{ \boldsymbol{v} \in \mathbb{Q}^m : \boldsymbol{v} = \sum_{i=1}^{n} \lambda_i \cdot \boldsymbol{v}_i, \lambda_i \in \mathbb{Q}, \lambda_i \geq 0 \right\}.$$

Note that for subsequent technical convenience, in our definition of the convex hull we require the λ_i to only sum up to less or equal to one, as opposed to exactly one which is more commonly found in the literature. We may do so as it is not difficult to check that for V as above there are $\boldsymbol{u} \in \mathbb{Q}^m$ and $U = \{\boldsymbol{u}_2, \ldots, \boldsymbol{u}_n\} \subseteq \mathbb{Q}^m$ such that $\|\boldsymbol{u}\|, \|U\| \leq O(\|V\|)$ and

$$\boldsymbol{u} + \operatorname{conv} U = \left\{ \boldsymbol{v} \in \mathbb{Q}^m : \boldsymbol{v} = \sum_{i=1}^{n} \lambda_i \cdot \boldsymbol{v}_i, \lambda_i \in \mathbb{Q}, 0 \leq \lambda_i \leq 1, \sum_{i=1}^{n} \lambda_i = 1 \right\},$$

and *vice versa*.

We first recall a classical result from the theory of linear programming that states that the set of solutions of a systems of linear inequalities can be obtained as the sum of a finite polyhedron and a cone.

PROPOSITION 4.1. *[Schrijver 1998, Thm. 10.2] Let $S : A \cdot \boldsymbol{x} \geq \boldsymbol{c}$ be a system of linear inequalities. Then there exist a rational vector $\boldsymbol{u}$ and finite sets of rational vectors $\{\boldsymbol{v}_j\}_{j \in J}$, $\{\boldsymbol{w}_k\}_{k \in K}$ such that $\langle \boldsymbol{u} \rangle, \langle \boldsymbol{v}_j \rangle, \langle \boldsymbol{w}_k \rangle \leq \mathsf{poly}(\langle S \rangle)$ and*

$$[\![S]\!] = \boldsymbol{u} + \operatorname{conv}\{\boldsymbol{v}_j\}_{j \in J} + \operatorname{cone}\{\boldsymbol{w}_k\}_{k \in K}.$$

In our first result, we apply Proposition 4.1 in order to show that non-empty systems of mixed linear inequalities and strict inequalities contain points whose representation size is polynomial in the facet complexity of the system. For such a system $S : A \cdot \boldsymbol{x} \geq \boldsymbol{c} \wedge B \cdot \boldsymbol{x} > \boldsymbol{d}$, by $\overline{S}$ we denote the system of linear inequalities $\overline{S} : \left(\begin{smallmatrix} A \\ B \end{smallmatrix}\right) \cdot \boldsymbol{x} \geq \left(\begin{smallmatrix} \boldsymbol{c} \\ \boldsymbol{d} \end{smallmatrix}\right)$, whose set of solutions is the topological closure of the solutions of S.

PROPOSITION 4.2. *Let $S : A \cdot \boldsymbol{x} \geq \boldsymbol{c} \wedge B \cdot \boldsymbol{x} > \boldsymbol{d}$ be a system of mixed linear inequalities and strict inequalities such that $[\![S]\!] \neq \emptyset$. Then there is $\boldsymbol{v} \in [\![S]\!]$ such that $\langle \boldsymbol{v} \rangle \leq \mathsf{poly}(\langle S \rangle)$.*

PROOF. Let A and B be $m \times n$ and $m' \times n$ matrices, respectively. Let $\boldsymbol{z} \in [\![S]\!]$, and consequently $\boldsymbol{z} \in [\![\overline{S}]\!]$. By Proposition 4.1,

$$[\![\overline{S}]\!] = \boldsymbol{u} + \operatorname{conv}\{\boldsymbol{v}_j\}_{j \in J} + \operatorname{cone}\{\boldsymbol{w}_k\}_{k \in K}$$

for some index sets J, K, and vectors $\boldsymbol{u}$, $\{\boldsymbol{v}_j\}_{j \in J}$ and $\{\boldsymbol{w}_k\}_{k \in K}$ of bit size polynomial in $\langle \overline{S} \rangle = \langle S \rangle$.

Let E be the linear subspace generated by $\{\boldsymbol{v}_j\}_{j\in J}$ and $\{\boldsymbol{w}_k\}_{k\in K}$. It has dimension at most n. So there are subsets $\{\boldsymbol{v}_1,\dots \boldsymbol{v}_r\} \subseteq \{\boldsymbol{v}_j\}_{j\in J}$ and some $\{\boldsymbol{w}_1,\dots \boldsymbol{w}_s\} \subseteq \{\boldsymbol{w}_k\}_{k\in K}$ such that $\{\boldsymbol{v}_1,\dots \boldsymbol{v}_r, \boldsymbol{w}_1,\dots \boldsymbol{w}_s\}$ generates E, $r > 0$ and $r + s \leq n$. In particular, $\boldsymbol{z} = \boldsymbol{u} + \sum_{1\leq j\leq r} \lambda_j \cdot \boldsymbol{v}_j + \sum_{1\leq j\leq s} \mu_j \cdot \boldsymbol{w}_j$ for some $\{\lambda_j\}_{1\leq j\leq r}, \{\mu_j\}_{1\leq j\leq s} \subseteq \mathbb{Q}$.

Now define $\boldsymbol{z}' \stackrel{\text{def}}{=} \boldsymbol{u} + \sum_{1\leq j\leq r} \frac{1}{2\cdot r} \cdot \boldsymbol{v}_j + \sum_{1\leq j\leq s} \boldsymbol{w}_j$. By construction $\boldsymbol{z}' \in \overline{S}$ and $\langle \boldsymbol{z}'\rangle \leq \mathsf{poly}(\langle S\rangle)$. We claim that $\boldsymbol{z}' \in S$. For the sake of contradiction, assume that some strict inequality of $B \cdot \boldsymbol{z}' > \boldsymbol{d}$ is violated, say $\boldsymbol{b} \cdot \boldsymbol{z}' = d$ for some row $\boldsymbol{b} \cdot \boldsymbol{x} > d$ of $B \cdot \boldsymbol{x} > d$. For some $\varepsilon > 0$, consider vector $\boldsymbol{z}_\varepsilon = (1 + \varepsilon) \cdot \boldsymbol{z}' - \varepsilon \cdot \boldsymbol{z}$. On the one hand, we have

$$\boldsymbol{b} \cdot \boldsymbol{z}_\varepsilon = (1 + \varepsilon) \cdot \boldsymbol{b} \cdot \boldsymbol{z}' - \varepsilon \cdot \boldsymbol{b} \cdot \boldsymbol{z} = d - \varepsilon \cdot (\boldsymbol{b} \cdot \boldsymbol{z} - d) < d.$$

On the other hand, $\boldsymbol{z}_\varepsilon = \boldsymbol{u} + \sum_{1\leq j\leq r} (\frac{1+\varepsilon}{2\cdot r} - \varepsilon \cdot \lambda_j) \cdot \boldsymbol{v}_j + \sum_{1\leq j\leq s} (1 + \varepsilon - \varepsilon \cdot \mu_j) \cdot \boldsymbol{w}_j$. Thus for ε enough small, $\boldsymbol{z}_\varepsilon \in \overline{S}$ which yields a contradiction. □

Next, we recall the reverse direction of Proposition 4.1.

PROPOSITION 4.3. *[Schrijver 1998, Thm. 10.2] Let $\boldsymbol{u}$ be a rational vector, and let $\{\boldsymbol{v}_j\}_{j\in J}$, $\{\boldsymbol{w}_k\}_{k\in K}$ be finite sets of rational vectors. There exists a system of linear inequalities $S : A \cdot \boldsymbol{x} \geq \boldsymbol{b}$ such that $\langle S\rangle \leq \mathsf{poly}(\langle \boldsymbol{u}\rangle + \max_{j\in J}\langle \boldsymbol{v}_j\rangle + \max_{k\in K}\langle \boldsymbol{w}_k\rangle)$ and*

$$[\![S]\!] = \boldsymbol{u} + \mathrm{conv}\{\boldsymbol{v}_j\}_{j\in J} + \mathrm{cone}\{\boldsymbol{w}_k\}_{k\in K}.$$

We now turn towards projections of systems of mixed linear inequalities and strict inequalities. Given such a system $S : A \cdot \boldsymbol{x} \geq \boldsymbol{c} \wedge B \cdot \boldsymbol{x} > \boldsymbol{d}$ such that $\boldsymbol{x} = (\boldsymbol{y}, \boldsymbol{z})$, we denote by $\pi_{\boldsymbol{y}}[\![S]\!]$ the projection of the set of solutions of S onto $\boldsymbol{y}$. The next proposition establishes that projections of the set of solutions of systems of mixed linear inequalities and strict inequalities can be obtained as the union of the set of solutions of a finite number of systems of mixed linear equalities and strict inequalities.

PROPOSITION 4.4. *Let $S : A\cdot\boldsymbol{x} \geq \boldsymbol{c}\wedge B\cdot\boldsymbol{x} > \boldsymbol{d}$ be a system of mixed linear inequalities and strict inequalities such that $\boldsymbol{x} = (\boldsymbol{y}, \boldsymbol{z})$. Then there are a finite number of systems of mixed linear equalities and strict inequalities $S_i : A_i \cdot \boldsymbol{y} = \boldsymbol{b}_i \wedge B_i \cdot \boldsymbol{y} > \boldsymbol{d}_i$, $i \in I$, such that $\langle S_i\rangle \leq \mathsf{poly}(\langle S\rangle)$ for all $i \in I$, and*

$$\pi_{\boldsymbol{y}}[\![S]\!] = \bigcup_{i\in I} [\![S_i]\!].$$

PROOF. By Proposition 4.1, there exist a rational vector $\boldsymbol{u}$ and finite sets of rational vectors $\{\boldsymbol{v}_j\}_{j\in J}$, $\{\boldsymbol{w}_k\}_{k\in K}$ for some index sets J, K such that $\langle \boldsymbol{u}\rangle, \langle \boldsymbol{v}_j\rangle, \langle \boldsymbol{w}_k\rangle \leq \mathsf{poly}(\langle S\rangle)$ and

$$[\![\overline{S}]\!] = \boldsymbol{u} + \mathrm{conv}\{\boldsymbol{v}_j\}_{j\in J} + \mathrm{cone}\{\boldsymbol{w}_k\}_{k\in K}.$$

Projecting all vectors onto the $\boldsymbol{y}$-components and applying Proposition 4.3, we obtain a system of linear inequalities $S' : C \cdot \boldsymbol{y} \geq \boldsymbol{e}$ such that $\pi_{\boldsymbol{y}}[\![\overline{S}]\!] = [\![S']\!]$ and $\langle S'\rangle \leq \mathsf{poly}(\langle S\rangle)$. Write S' as a conjunction of linear inequalities $\Psi_{S'}$ such that $\Psi_{S'} = \bigwedge_{k\in K} \boldsymbol{c}_k \cdot \boldsymbol{y} \geq e_k$. For a partition of K as $K = E \uplus G$, let $S_{E,G}$ be the system of mixed linear equalities and strict inequalities corresponding to $\Psi_{E,G} : \bigwedge_{k\in E} \boldsymbol{c}_k \cdot \boldsymbol{y} = e_k \wedge \bigwedge_{k\in G} \boldsymbol{c}_k \cdot \boldsymbol{y} > e_k$. Moreover, let

$$I \stackrel{\text{def}}{=} \{(E, G) : \pi_{\boldsymbol{y}}[\![S]\!] \cap [\![S_{E,G}]\!] \neq \emptyset\}.$$

We claim that $\pi_{\boldsymbol{y}}[\![S]\!] = \bigcup_{i\in I} [\![S_i]\!]$. Since $\pi_{\boldsymbol{y}}[\![S]\!] \subseteq [\![S']\!]$, we have

$$\pi_{\boldsymbol{y}}[\![S]\!] = \pi_{\boldsymbol{y}}[\![S]\!] \cap [\![S']\!] = \bigcup_{K=E\uplus G} \pi_{\boldsymbol{y}}[\![S]\!] \cap [\![S_{E,G}]\!] = \bigcup_{i\in I} \pi_{\boldsymbol{y}}[\![S]\!] \cap [\![S_i]\!] \subseteq \bigcup_{i\in I} [\![S_i]\!].$$

It thus remains to show that $[\![S_{E,G}]\!] \subseteq \pi_{\boldsymbol{y}}[\![S]\!]$ for every $(E,G) \in I$. To this end, let $(E,G) \in I$, let $\boldsymbol{u} \in \pi_{\boldsymbol{y}}[\![S]\!] \cap [\![S_{E,G}]\!]$ and let $\boldsymbol{v} \in [\![S_{E,G}]\!]$. For $\varepsilon > 0$, set $\boldsymbol{v}_\varepsilon \stackrel{\text{def}}{=} (1+\varepsilon)\cdot\boldsymbol{v} - \varepsilon\cdot\boldsymbol{u}$. For all $k \in E$ we have $\boldsymbol{c}_k \cdot \boldsymbol{v}_\varepsilon = e_k$, and for all $k \in G$ we have $\boldsymbol{c}_k \cdot \boldsymbol{v}_\varepsilon = \boldsymbol{c}_k \cdot \boldsymbol{v} + \varepsilon \cdot \boldsymbol{c}_k \cdot (\boldsymbol{v} - \boldsymbol{u})$. Thus, for small enough ε, we have $\boldsymbol{v}_\varepsilon \in [\![S_{E,G}]\!]$. Since $\boldsymbol{u} \in \pi_{\boldsymbol{y}}[\![S]\!]$ and $\boldsymbol{v}_\varepsilon \in [\![S_{E,G}]\!] \subseteq [\![S']\!] = \pi_{\boldsymbol{y}}[\![\overline{S}]\!]$, there exist $\boldsymbol{w}, \boldsymbol{w}_\varepsilon$ such that $(\boldsymbol{u},\boldsymbol{w}) \in [\![S]\!]$ and $(\boldsymbol{v}_\varepsilon, \boldsymbol{w}_\varepsilon) \in [\![\overline{S}]\!]$. It can easily be shown that $\alpha_1 \cdot \boldsymbol{z}_1 + \alpha_2 \cdot \boldsymbol{z}_2 \in [\![S]\!]$ for every $\boldsymbol{z}_1 \in [\![S]\!]$, $\boldsymbol{z}_2 \in [\![\overline{S}]\!]$ and $\alpha_1, \alpha_2 > 0$ such that $\alpha_1 + \alpha_2 = 1$. Therefore, in particular, $\frac{\varepsilon}{\varepsilon+1} \cdot (\boldsymbol{u},\boldsymbol{w}) + \frac{1}{\varepsilon+1} \cdot (\boldsymbol{v}_\varepsilon, \boldsymbol{w}_\varepsilon) \in [\![S]\!]$. Now, observe that $\boldsymbol{v} = \frac{\varepsilon}{1+\varepsilon}\boldsymbol{u} + \frac{1}{1+\varepsilon}\boldsymbol{v}_\varepsilon$. This implies that $(\boldsymbol{v}, \frac{\varepsilon}{1+\varepsilon} \cdot \boldsymbol{w} + \frac{1}{1+\varepsilon} \cdot \boldsymbol{w}_\varepsilon) \in [\![S]\!]$, and hence that $\boldsymbol{v} \in \pi_{\boldsymbol{y}}[\![S]\!]$. □

Finally, we can employ Proposition 4.4 in order to provide a quantifier elimination procedure for FO$\langle\mathbb{Q},+,<\rangle$ which yields formulas of small facet complexity. Given a quantifier-free formula $\varphi(\boldsymbol{x})$, recall that we may assume that no negation symbols occur in $\varphi(\boldsymbol{x})$, and that the only relation symbols occurring in $\varphi(\boldsymbol{x})$ are $>$ and $\geq$. Hence, the disjunctive normal form of $\varphi(\boldsymbol{x})$ can be written as

$$\varphi(\boldsymbol{x}) \equiv \bigvee_{i \in I} A_i \cdot \boldsymbol{x} \geq \boldsymbol{c}_i \wedge B_i \cdot \boldsymbol{x} > \boldsymbol{d}_i.$$

We define the facet complexity $\langle\varphi(\boldsymbol{x})\rangle$ as the maximum over $\langle S_i\rangle$, $i \in I$, where each S_i is $A_i \cdot \boldsymbol{x} \geq \boldsymbol{c}_i \wedge B_i \cdot \boldsymbol{x} > \boldsymbol{d}_i$. Observe that negating $\varphi(\boldsymbol{x})$ does not change its facet complexity.

COROLLARY 4.5. *Let $\varphi(\boldsymbol{y},\boldsymbol{z})$ be a quantifier-free FO$\langle\mathbb{Q},+,<\rangle$-formula. Then there is $\psi(\boldsymbol{y})$ such that $[\![\psi(\boldsymbol{y})]\!] = \pi_{\boldsymbol{y}}[\![\varphi(\boldsymbol{y},\boldsymbol{z})]\!]$ and $\langle\psi(\boldsymbol{y})\rangle \leq \mathsf{poly}(\langle\varphi(\boldsymbol{y},\boldsymbol{z})\rangle)$.*

PROOF. Let $\boldsymbol{x} = (\boldsymbol{y},\boldsymbol{z})$, and let $S_i : A_i \cdot \boldsymbol{x} \geq \boldsymbol{c}_i \wedge B_i \cdot \boldsymbol{x} > \boldsymbol{d}_i$, $i \in I$, be all systems of mixed linear inequalities and strict inequalities in the disjunctive normal form of φ. By Proposition 4.4, for every S_i there exist systems $S_{i,j} : A_{i,j} \cdot \boldsymbol{y} \geq \boldsymbol{c}_{i,j} \wedge B_{i,j} \cdot \boldsymbol{y} > \boldsymbol{d}_{i,j}$, $j \in J_i$, such that $\pi_{\boldsymbol{y}}[\![S_i]\!] = \bigcup_{j \in J_i} [\![S_{i,j}]\!]$, and $\langle S_{i,j}\rangle \leq \mathsf{poly}(\langle S_i\rangle)$. Consequently, $\psi(\boldsymbol{y}) \stackrel{\text{def}}{=} \bigvee_{i \in I} \bigvee_{j \in J_i} A_{i,j} \cdot \boldsymbol{y} \geq \boldsymbol{c}_{i,j} \wedge B_{i,j} \cdot \boldsymbol{y} > \boldsymbol{d}_{i,j}$ has the desired properties. □

4.2. The Inclusion Problem

Given continuous Petri nets $\mathcal{N}, \mathcal{M}$ with the same set of places and initial markings $\boldsymbol{m}, \boldsymbol{m}'$, inclusion asks whether every marking $\boldsymbol{m}''$ reachable from $\boldsymbol{m}$ in $\mathcal{N}$ is also reachable in $\mathcal{M}$ from $\boldsymbol{m}'$. Recall the logical definition of inclusion provided in Table I given as $\Phi_{inc}^{\mathcal{N},\mathcal{M}}(\boldsymbol{x},\boldsymbol{x}') \stackrel{\text{def}}{=} \forall \boldsymbol{x}'' : \Phi_{\mathcal{N}}(\boldsymbol{x},\boldsymbol{x}'') \rightarrow \Phi_{\mathcal{M}}(\boldsymbol{x}',\boldsymbol{x}'')$. Here, we improve the EXP upper bound developed by Fraca and Haddad [2015] and the immediate Π_2^{P}-upper bound from Proposition 2.5, and show that the coNP lower bound given by Fraca and Haddad [2015] is actually tight.

PROPOSITION 4.6. *Inclusion for continuous Petri nets is* coNP*-complete.*

PROOF. For fixed initial markings $\boldsymbol{m}$, $\boldsymbol{m}'$, using standard transformations and renaming of variables, we have that $\Phi_{inc}^{\mathcal{N},\mathcal{M}}(\boldsymbol{m},\boldsymbol{m}')$ is equivalent to a Π_2-sentence $\varphi = \forall \boldsymbol{x} : \exists \boldsymbol{y} : \psi(\boldsymbol{x},\boldsymbol{y})$ such that $\langle\psi(\boldsymbol{x},\boldsymbol{y})\rangle \leq |\Phi_{inc}^{\mathcal{N},\mathcal{M}}(\boldsymbol{m},\boldsymbol{m}')| \leq \mathsf{poly}(|\mathcal{M}| + |\mathcal{N}| + \langle\boldsymbol{m}\rangle + \langle\boldsymbol{m}'\rangle)$. By Corollary 4.5, there is $\psi'(\boldsymbol{x})$ such that $\varphi \equiv \forall \boldsymbol{x} : \psi'(\boldsymbol{x})$ and $\langle\psi'(\boldsymbol{x})\rangle \leq \mathsf{poly}\langle\psi(\boldsymbol{x},\boldsymbol{y})\rangle$. Hence, if φ is invalid it follows from Proposition 4.2 that there exists some witness $\boldsymbol{v}$ such that $\psi'(\boldsymbol{v})$ does not hold and $\langle\boldsymbol{v}\rangle \leq \mathsf{poly}(\langle\psi'(\boldsymbol{x})\rangle)$.

Consequently, an NP algorithm deciding non-inclusion proceeds as follows: guess a $\boldsymbol{v}$ such that $\langle\boldsymbol{v}\rangle \leq \mathsf{poly}(|\mathcal{M}|+|\mathcal{N}|+\langle\boldsymbol{m}\rangle+\langle\boldsymbol{m}'\rangle)$, and by Proposition 2.4 verify in polynomial time that $(\boldsymbol{m},\boldsymbol{v}) \in \mathbb{Q}\mathcal{R}(\mathcal{N})$ and $(\boldsymbol{m},\boldsymbol{v}) \notin \mathbb{Q}\mathcal{R}(\mathcal{M})$. Hence, inclusion for continuous Petri nets is in coNP. □

Remark 4.7. It is worth mentioning that Proposition 4.6 additionally yields a coNP-completeness result for reversibility of a continuous Petri net. Given a Petri net $\mathcal{N}$ and a configuration $\boldsymbol{m}$, reversibility is to decide whether $\mathbb{Q}\mathcal{R}(\mathcal{N})(\boldsymbol{m}) \subseteq \mathbb{Q}\mathcal{R}(\mathcal{N}^{-1})(\boldsymbol{m})$ holds. This problem is coNP-hard [Fraca and Haddad 2015] and by Proposition 4.6 in coNP, hence coNP-complete.

4.3. ε-Liveness Problems

Given a continuous Petri net $\mathcal{N}$ and a marking $\boldsymbol{m}$, recall that the ε-liveness problem asks whether there exists an $\varepsilon > 0$ such that for every marking $\boldsymbol{m}'$ reachable from $\boldsymbol{m}$ and every transition $t \in T$, a marking $\boldsymbol{m}''$ is reachable such that t has enabling degree at least ε in $\boldsymbol{m}''$. Recall the logical definition from Table I:

$$\Phi^{\mathcal{N}}_{lv}(\boldsymbol{x}) \stackrel{\text{def}}{=} \exists \varepsilon : \varepsilon > 0 \wedge \forall \boldsymbol{x}' : \Phi_{\mathcal{N}}(\boldsymbol{x}, \boldsymbol{x}') \to \bigwedge_{t \in T} \left(\exists \boldsymbol{x}'' : \Phi_{\mathcal{N}}(\boldsymbol{x}', \boldsymbol{x}'') \wedge \Phi^{\mathcal{N},t}_{nbld}(\varepsilon, \boldsymbol{x}'') \right).$$

Structural ε-liveness asks whether there exists some initial marking $\boldsymbol{m}$ such that $\mathcal{N}$ is ε-live in $\boldsymbol{m}$, i.e., whether $\Phi^{\mathcal{N}}_{slv} \stackrel{\text{def}}{=} \exists \boldsymbol{x} : \Phi^{\mathcal{N}}_{lv}(\boldsymbol{x})$ is valid. To the best of our knowledge, the following is the first decidability and complexity result for (structural) ε-liveness.

PROPOSITION 4.8. *ε-Liveness and structural ε-liveness are decidable in Σ^{P}_3.*

PROOF. Inspecting $\Phi^{\mathcal{N}}_{slv}$, we see that it is a Σ_3-sentence in FO$\langle \mathbb{Q}, +, < \rangle$. Consequently, an application of Proposition 2.5 yields the desired upper bounds. □

Unfortunately, there does not seem to be an obvious way to decrease those upper bounds, though we suspect them not to be tight. The precise complexity of (structural) ε-liveness remains an open problem of this paper.

4.4. Home-State Problems

Given a continuous Petri net $\mathcal{N}$ and an initial marking $\boldsymbol{m}$, a marking $\boldsymbol{m}'$ is a home state if $\boldsymbol{m}'$ can be reached from every marking $\boldsymbol{m}''$ that can be reached from $\boldsymbol{m}$. The existential home-state problem is to decide whether there exists a home state for a given initial marking $\boldsymbol{m}$. Again, we recall the logical definitions of those problems from Table I: $\Phi^{\mathcal{N}}_{hm}(\boldsymbol{x}, \boldsymbol{x}') \stackrel{\text{def}}{=} \forall \boldsymbol{x}'' : \Phi_{\mathcal{N}}(\boldsymbol{x}, \boldsymbol{x}'') \to \Phi_{\mathcal{N}}(\boldsymbol{x}'', \boldsymbol{x}')$ for the home-state problem, and $\Phi^{\mathcal{N}}_{ehm}(\boldsymbol{x}) \stackrel{\text{def}}{=} \exists \boldsymbol{x}' : \Phi_{hm}(\boldsymbol{x}, \boldsymbol{x}')$ for the existential home-state problem.

PROPOSITION 4.9. *The home-state problem for continuous Petri nets is* coNP*-complete.*

PROOF. The home-state problem can be rephrased as an inclusion problem. We have that $\boldsymbol{m}'$ is a home state if and only if $\mathbb{Q}\mathcal{R}(\mathcal{N})(\boldsymbol{m}) \subseteq \mathbb{Q}\mathcal{R}(\mathcal{N}^{-1})(\boldsymbol{m}')$. Consequently, by Proposition 4.6, it is decidable in coNP.

To show coNP-hardness of the home-state problem, we adapt a reduction from 3-SAT used by Fraca and Haddad [2015, Prop. 34] and Desel and Esparza [1995, Thm. 4.28] to show that the liveness problem is coNP-complete, respectively, for continuous and discrete free-choice Petri nets. Let φ be a formula in 3-CNF with k variables $x_1, x_2, \ldots, x_k$ and m clauses $C_1, C_2, \ldots, C_m$. For each clause C_j, we define $\ell_{j,1}, \ell_{j,2}$ and $\ell_{j,3}$ as its three literals. Consider the continuous Petri net $\mathcal{N}$ obtained from φ as follows:

— for every $1 \leq i \leq k$, we add a place p_i initially marked by the marking $\boldsymbol{m}_0$, and transitions t_i and f_i that both have p_i as an their sole input place;
— for every $1 \leq j \leq m$, we add empty places $q_{j,1}, q_{j,2}, q_{j,3}$ such that $q_{j,b}$ is an output place of t_i if $q_{j,b} = \neg x_i$ or an output place of f_i if $q_{j,b} = x_i$;
— for every $1 \leq j \leq m$, we add a transition c_j with $q_{j,1}$, $q_{j,2}$ and $q_{j,3}$ as its only input places.

— we add an empty place p_{main} as the output place of $c_1, c_2, \ldots, c_m$.

It was shown by Fraca and Haddad [2015, Prop. 34] that φ is unsatisfiable if, and only if, for every marking $\boldsymbol{m}$ reachable from $\boldsymbol{m}_0$ in $\mathcal{N}$, there exists a marking $\boldsymbol{m}'$ reachable from $\boldsymbol{m}$ such that $\boldsymbol{m}'(p_{main}) > 0$.

We exploit this observation to obtain a reduction to the home-state problem. Let $\mathcal{N}'$ be the continuous Petri net obtained from $\mathcal{N}$ as follows:

— we add a transition t_{dec} that removes two tokens from p_{main} and adds one token to p_{main};
— we add a transition t_{inc} that removes one token from p_{main} and adds two token to p_{main};
— for every place $p \neq p_{main}$ from $\mathcal{N}$, we add a transition t_p with input places p and p_{main}, and output place p_{main}.

Let $\boldsymbol{m}$ be a marking such that $\boldsymbol{m}(p_{main}) > 0$. It is not so difficult to see that it is possible to increase or decrease $\boldsymbol{m}(p_{main})$ to any positive value, but never to 0; in particular, a marking of p_{main} with value 1 can always be reached. Moreover, from $\boldsymbol{m}$, any place $p \neq p_{main}$ can reach zero by executing t_p by the appropriate amount.

Therefore, if φ is unsatisfiable, by the above observation on $\mathcal{N}$, any marking reachable from $\boldsymbol{m}_0$ can reach a marking with a non-zero value in p_{main}. Moreover, by construction of $\mathcal{N}'$, any marking with a non-zero value in p_{main} can lead to a marking where all of the other places are emptied, and then where p_{main} is set to 1.

Let $\boldsymbol{m}_{home}$ be the marking of $\mathcal{N}'$ defined by $\boldsymbol{m}_{home}(p) = 1$ if $p = p_{main}$ and 0 otherwise. We conclude that φ is unsatisfiable if, and only if, $\boldsymbol{m}_{home}$ is a home state of $\mathcal{N}'$. □

PROPOSITION 4.10. *The existential home-state problem is decidable in* Σ_2^{P}.

PROOF. For a fixed initial marking $\boldsymbol{m}$, by repeated quantifier elimination we have that $\Phi_{ehm}^{\mathcal{N}}(\boldsymbol{m}) \equiv \exists \boldsymbol{x} : \psi(\boldsymbol{x})$ for some $\psi(\boldsymbol{x})$ such that $\langle \psi(\boldsymbol{x}) \rangle \leq \mathsf{poly}(\langle \Phi_{ehm}^{\mathcal{N}}(\boldsymbol{m}) \rangle) \leq \mathsf{poly}(|\mathcal{N}| + \langle \boldsymbol{m} \rangle)$. Consequently, it follows from Proposition 4.2 that if there exists a home state $\boldsymbol{m}'$ then there is one, say $\boldsymbol{v}$, such that $\langle \boldsymbol{v} \rangle \leq \mathsf{poly}(|\mathcal{N}| + \langle \boldsymbol{m} \rangle)$. Hence, $\boldsymbol{v}$ can be guessed in NP, and using a coNP oracle it can be verified that $\boldsymbol{v}$ is indeed a home state with respect to the initial marking $\boldsymbol{m}$. □

As remarked by Jančar [2017], structural versions of Petri-net decision problems are often easier to decide, as is exemplified by the structural boundedness problem for discrete Petri nets. This problem is solvable in P, see e.g. [Thoen and Catthoor 2000], whereas boundedness is EXPSPACE-complete [Lipton 1976; Rackoff 1978]. It does not seem inconceivable that the existential home-state problem can be decided with lower complexity.

5. COVERABILITY IN PETRI NETS VIA REACHABILITY IN CONTINUOUS PETRI NETS

Recall that, given a (continuous or discrete) Petri net $\mathcal{N}$, an initial marking $\boldsymbol{m}_0$ and a target marking $\boldsymbol{m}$, the coverability problem asks whether $\boldsymbol{m}$ is *coverable* from $\boldsymbol{m}_0$ in $\mathcal{N}$, i.e., whether $\boldsymbol{m}_0 \to^* \boldsymbol{m}'$ for some marking $\boldsymbol{m}' \geq \boldsymbol{m}$.

The coverability problem was one of the first decision problems shown decidable for discrete Petri nets [Karp and Miller 1967; Hack 1976]. In the approach pioneered by Karp and Miller, a finite tree representing markings coverable from $\boldsymbol{m}_0$ is constructed. This tree is obtained from building the reachability tree starting from the initial marking $\boldsymbol{m}_0$ and computing so-called accelerations whenever a node has a smaller or equal ancestor. The scalability of this approach is limited since the size of this tree may be non-primitive recursive. Nevertheless, heuristics have been investigated that keep the

size of the tree manageable in practice, see e.g. [Geeraerts et al. 2010; Reynier and Servais 2013; Valmari and Hansen 2014].

Another popular approach for solving the coverability problem, namely the *backward approach*, that forms the basis of the approach that we present in this paper, was introduced by Arnold and Latteux [1978] for vector addition systems with resets, and first formalized and popularized by Abdulla et al. [1996] in the more general context of well-structured transition systems. This approach is based on an algorithm that begins with the target marking $\boldsymbol{m}$ and iteratively computes predecessors of $\boldsymbol{m}$ from which a marking larger or equal to $\boldsymbol{m}$ may be reached; hence its name, the *backward algorithm*. A bottleneck of the backward algorithm is that its number of iterations may be doubly exponential in the worst case [Bozzelli and Ganty 2011]. In particular, as the set of predecessors computed by the algorithm may also grow doubly exponentially [Bozzelli and Ganty 2011], computations tend to become much slower as the number of iterations increases, even on Petri nets of relatively modest size.

In this section, we show how to speedup computations of the backward algorithm by exploiting reachability in continuous Petri nets as a pruning criterion. That is, in every iteration of the backward algorithm, we use the over approximation provided by continuous Petri nets in order to discard all predecessors which define a set of configurations which are not coverable in the continuous semantics. A cornerstone of the empirical performance of our approach is that the logical characterization of reachability in continuous Petri nets enables us to employ SMT-solvers in our prototype implementation for those purposes.

This section is structured as follows. In Section 5.1, we introduce some auxiliary definitions and recall the backward algorithm. Subsequently, in Section 5.2 we develop our variant of the backward algorithm in which $\mathbb{Q}$-coverability is used as a pruning criterion. Section 5.3 describes an implementation and the evaluation of the algorithm developed in Section 5.2. Finally, in Section 5.4 we discuss the relationship of our variant of the backward algorithm to approaches that have appeared in the literature.

5.1. The Backward Algorithm

We first present the classical backward algorithm and introduce some auxiliary definitions. A set $V \subseteq \mathbb{N}^P$ is *upward-closed* if for every $\boldsymbol{v} \in V$ and $\boldsymbol{w} \in \mathbb{N}^P$, $\boldsymbol{v} \leq \boldsymbol{w}$ implies $\boldsymbol{w} \in V$. The *upward closure* of a vector $\boldsymbol{v} \in \mathbb{N}^P$ is the set

$$\uparrow\boldsymbol{v} \stackrel{\text{def}}{=} \{\boldsymbol{w} \in \mathbb{N}^P : \boldsymbol{v} \leq \boldsymbol{w}\}.$$

This definition can be lifted to sets $V \subseteq \mathbb{N}^P$ in the obvious way, i.e., $\uparrow V \stackrel{\text{def}}{=} \bigcup_{\boldsymbol{v} \in V} \uparrow\boldsymbol{v}$. Due to $\mathbb{N}^P$ being well-quasi-ordered by $\leq$, any upward-closed set V contains a finite subset $F \subseteq V$ such that $V = \uparrow F$. Such an F is called a *basis* of V and allows for a finite representation of an upward-closed set. In particular, it can be shown that V contains a unique *minimal basis* $B \subseteq V$ that is minimal with respect to inclusion for all bases $F \subseteq V$. We use $minbase(V)$ to denote this minimal basis. For any finite basis F of V, $minbase(V)$ is obtained by deleting vectors $\boldsymbol{v} \in F$ such that there exists $\boldsymbol{w} \in F$ with $\boldsymbol{w} < \boldsymbol{v}$. For a Petri net $\mathcal{N} = (P, T, \mathbf{Pre}, \mathbf{Post})$ and a set of markings M, we define $pred(M)$ as the set of predecessors of M, i.e., $pred(M) \stackrel{\text{def}}{=} \{\boldsymbol{m} \in \mathbb{N}^P : \boldsymbol{m} \xrightarrow{t} \boldsymbol{m}' \text{ for some } t \in T, \boldsymbol{m}' \in M\}$. Note that for every upward-closed set M, $pred(M)$ is upward-closed.

Formally, the backward algorithm computes a sequence

$$M_0, M_1, M_2, \ldots \tag{3}$$

such that $\uparrow M_0 = \uparrow\boldsymbol{m}$ and $\uparrow M_i = pred(\uparrow M_{i-1})$ for every $i > 0$. Since Petri nets are well-structured, (3) stabilizes to some M_n such that $\uparrow M_n = \{\boldsymbol{x} \in \mathbb{N}^P : \boldsymbol{x} \rightarrow^*$

ALGORITHM 1: Backward algorithm solving the coverability problem for Petri nets.

Input: Petri net $\mathcal{N} = (P, T, \mathbf{Pre}, \mathbf{Post})$ and $\boldsymbol{m}_0, \boldsymbol{m} \in \mathbb{N}^P$.
Output: Does there exist $\boldsymbol{m}' \in \uparrow\boldsymbol{m}$ such that $\boldsymbol{m}_0 \to^* \boldsymbol{m}'$?

```
1  M ← {m}
2  while m_0 ∉ ↑M do
3  |  B ← pb(M) \ ↑M
4  |  if B = ∅ then
5  |  |  return false
6  |  else
7  |  |  M ← minbase(M ∪ B)
8  |  end
9  end
10 return true
```

$\boldsymbol{m}'$ for some $\boldsymbol{m}' \geq \boldsymbol{m}\}$ (see e.g. [Abdulla et al. 1996; Finkel and Schnoebelen 2001]). Therefore, the target marking $\boldsymbol{m}$ is coverable from the initial marking $\boldsymbol{m}_0$ if, and only if, $\boldsymbol{m}_0 \in \uparrow M_n$.

We detail the backward algorithm in Algorithm 1. Each upward-closed set $\uparrow M_i$ is represented by its unique minimal basis M_i. Moreover, to implement the computation of $pred(\uparrow M_{i-1})$, we follow Finkel and Leroux [2015]. We associate to each marking $\boldsymbol{v} \in \mathbb{N}^P$ and each transition $t \in T$ the marking $\boldsymbol{v}_t$ defined by

$$\boldsymbol{v}_t(p) \stackrel{\text{def}}{=} \max\{\mathbf{Pre}(p,t), \boldsymbol{v}(p) - \mathbf{Incid}(p,t)\} \text{ for every } p \in P.$$

It is possible to verify that $\{\boldsymbol{v}_t\}$ is the minimal basis of markings covering $\boldsymbol{v}$ after firing t, i.e.

$$\{\boldsymbol{v}_t\} = minbase(\{\boldsymbol{u} \in \mathbb{N}^P : \text{there is } \boldsymbol{v}' \in \uparrow\boldsymbol{v} \text{ such that } \boldsymbol{u} \xrightarrow{t} \boldsymbol{v}'\}).$$

Let $V \subseteq \mathbb{N}^P$, we let $pb(V) \stackrel{\text{def}}{=} \bigcup_{\boldsymbol{u} \in V, t \in T} \{\boldsymbol{u}_t\}$, which allows us to obtain the potentially non-minimal finite basis $pb(V)$ of $pred(\uparrow V)$, i.e.

$$\begin{aligned} \uparrow pb(V) &= \{\boldsymbol{u} \in \mathbb{N}^P : \text{there is } \boldsymbol{v}' \in \uparrow V \text{ such that } \boldsymbol{u} \to \boldsymbol{v}'\} \\ &= pred(\uparrow V). \end{aligned}$$

5.2. The Backward Algorithm Modulo $\mathbb{Q}$-Coverability

We now present our extension of the classical backward algorithm that incorporates $\mathbb{Q}$-coverability checks during its execution in order to keep the set of minimal basis elements small. This algorithm is detailed in Algorithm 2. Blue font color indicates differences to Algorithm 1.

Let $\mathcal{N}$ be a Petri net, $\boldsymbol{m}_0$ an initial marking and $\boldsymbol{m}$ a target marking to cover. On Line 1, we first test whether $\boldsymbol{m}$ is $\mathbb{Q}$-coverable from $\boldsymbol{m}_0$. If it is the case, on Line 3, we derive an open formula $\psi(\boldsymbol{x})$ from $\Phi^{\mathcal{N}}_{cvr}(\boldsymbol{m}_0, \boldsymbol{x})$ from Table I such that $\psi(\boldsymbol{x})$ holds if and only if $\boldsymbol{x}$ is $\mathbb{Q}$-coverable in $\mathcal{N}$. On Lines 6 and 7, we prune new markings, i.e., markings from B that are not $\mathbb{Q}$-coverable from $\boldsymbol{m}_0$, and that would have thus never led to $\boldsymbol{m}_0$ in subsequent iterations.

PROPOSITION 5.1. *Let $\mathcal{N} = (P, T, \mathbf{Pre}, \mathbf{Post})$ be a Petri net and $\boldsymbol{m}_0, \boldsymbol{m} \in \mathbb{N}^P$. Algorithm 2 always halts, and returns "true" if, and only if, $\boldsymbol{m}$ is coverable from $\boldsymbol{m}_0$.*

ALGORITHM 2: Backward algorithm modulo $\mathbb{Q}$-coverability.

Input: Petri net $\mathcal{N} = (P, T, \mathbf{Pre}, \mathbf{Post})$ and $\boldsymbol{m}_0, \boldsymbol{m} \in \mathbb{N}^P$.
Output: Does there exist $\boldsymbol{m}' \in \uparrow\boldsymbol{m}$ such that $\boldsymbol{m}_0 \to^* \boldsymbol{m}'$?

```
1  if m is not Q-coverable from m_0 then return false
2  M ← {m}
3  ψ(x) ← Φ^N_cvr(m_0, x)
4  while m_0 ∉ ↑M do
5  |  B ← pb(M) \ ↑M
6  |  D ← {v ∈ B : ¬ψ(v)}
7  |  B ← B \ D
8  |  if B = ∅ then
9  |  |  return false
10 |  else
11 |  |  M ← minbase(M ∪ B)
12 |  end
13 end
14 return true
```

PROOF. Let B_n and M_n be respectively the values of B and M on Lines 7 and 11 of Algorithm 2 in the n-th iteration of the **while** loop. We note that

$$\uparrow M_n = \left\{ \boldsymbol{x} \in \mathbb{N}^P : \boldsymbol{m}_0 \to^*_{\mathbb{Q}} \boldsymbol{x'} \text{ and } \boldsymbol{x} \xrightarrow{k} \boldsymbol{y} \text{ for some } \boldsymbol{x}' \geq \boldsymbol{x}, \boldsymbol{y} \geq \boldsymbol{m} \text{ and } 0 \leq k \leq n \right\},$$
$$B_n \subseteq \uparrow M_n \setminus \uparrow M_{n-1}.$$

Moreover, we have that $\boldsymbol{m}$ is coverable from $\boldsymbol{m}_0$ if, and only if, $\boldsymbol{m}$ is $\mathbb{Q}$-coverable from $\boldsymbol{m}_0$ and there is $\boldsymbol{y} \in \uparrow\boldsymbol{m}$ such that $\boldsymbol{m}_0 \to^* \boldsymbol{y}$. Therefore, by definition of Lines 9 and 14, the algorithm is correct.

Since $\leq$ is a well-quasi-order for $\mathbb{N}^P$, we know that the sequence $\uparrow M_1 \subseteq \uparrow M_2 \subseteq \ldots$ stabilizes after a finite number of iterations n. Thus, after n iterations, either $B = \emptyset$ on Line 8, or $\boldsymbol{m}_0 \in \uparrow M$ on Line 4. In both cases, the algorithm halts. Therefore, it always halts. □

5.3. QCover: An Implementation of the Backward Algorithm Modulo $\mathbb{Q}$-Coverability

In this section, we provide an empirical evaluation of Algorithm 2 on a benchmark set from the literature. In Section 5.3.1, we discuss some details of our implementation, and in Section 5.3.2 we compare QCOVER against various tools from the literature.

5.3.1. Implementation Details. We have implemented the backward algorithm modulo $\mathbb{Q}$-coverability in a tool called QCOVER[2] in the programming language PYTHON. Petri Nets are represented by their **Pre** and **Post** matrices with the NUMPY[3] library. QCOVER also supports sparse matrices representation for very large Petri nets through SCIPY[4]. The input file format of coverability instances for QCOVER is a strict subset of the MIST file format[5].

In order to achieve a better performance, for a given coverability instance we first use a single pass of the polynomial-time algorithm of Fraca and Haddad [2015] in order to discharge instances which are not $\mathbb{Q}$-coverable. If the instance is $\mathbb{Q}$-coverable,

[2] QCOVER is available at https://github.com/blondimi/qcover/.
[3] NUMPY is available at http://www.numpy.org/.
[4] SCIPY is available at http://www.scipy.org/scipylib/.
[5] See https://github.com/pierreganty/mist/wiki#input-format-of-mist. Note that this format supports counter machines more general than Petri nets; QCOVER only supports a strict subset corresponding to Petri nets.

Table II. Left: number of safe and unsafe systems for each suite. Right: Petri nets average number of places and transitions for each suite.

Suite	Safe	Unsafe	Total
mist	23	4	27
bfc	2	44	46
soter	38	12	50
medical	12	0	12
bug_tracking	40	1	41
Total	115	61	176

Suite	Avg. num. of places	Avg. num. of transitions
mist	43	69
bfc	207	1,350
soter	2,805	4,985
medical	312	5,431
bug_tracking	754	27,370
Total	1,054	8,458

we resort to Algorithm 2. In order to check satisfiability of $\psi(\boldsymbol{x})$, we make use of the SMT solver Z3 [de Moura and Bjørner 2008], and we additionally interpret variables over $\mathbb{N}$ instead of $\mathbb{Q}_+$ with the goal of pruning more markings. The reason why we do not use the polynomial-time algorithm of Fraca and Haddad [2015] throughout the whole while-loop of Algorithm 2 is that on a single instance, this algorithm is usually faster, however when running multiple $\mathbb{Q}$-coverability queries on the *same* Petri net, caching strategies in Z3 yield a speed-up that cannot be achieved by any other means.

5.3.2. Benchmarks. We evaluated QCOVER on 176 systems modeled by Petri nets and coverability queries. This set of systems was used by Esparza et al. [2014] in order to benchmark the tool PETRINIZER presented therein, and is composed of the following five suites:

— `mist`: 27 systems drawn from the literature (mutual exclusion protocols, communication protocols, etc.) and used, in particular, to evaluate MIST[6];
— `bfc`: 46 systems obtained from concurrent C programs (multi-threaded programs with shared-memory, pseudorandom number generators, mutual exclusion protocols, etc.) and used, in particular, to test BFC[7] [Kaiser et al. 2012; Kaiser et al. 2014];
— `soter`: 50 systems obtained from concurrent ERLANG programs and used, in particular, by D'Osualdo et al. [2013] to test SOTER [D'Osualdo et al. 2012], a tool built on top of BFC;
— `medical`: 12 systems, described by Majumdar et al. [2013], modeling provenance analysis of messages of a simple medical messaging system of Vanderbilt University Medical Center;
— `bug_tracking`: 41 systems, described by Majumdar et al. [2013], modeling provenance analysis of messages of a bug-tracking system [Jank 2009].

As detailed in Table II, roughly two-thirds of the systems are safe (i.e., they are a no-instances of coverability). On average, the Petri nets used for the evaluation have 1054 places and 8458 transitions.

In order to evaluate our tool, we executed QCOVER and three other tools on the 176 systems with a timeout of 2000 secs. (33min and 20s) per instance. We compared QCOVER with the following tools: PETRINIZER [Esparza et al. 2014], MIST [Ganty 2002] and BFC [Kaiser et al. 2012; Kaiser et al. 2014] in their latest versions available at the time of writing [Blondin et al. 2016]. MIST implements a number of algorithms, we used the backward algorithm[8] that uses places invariant pruning [Ganty et al. 2007]. All benchmarks were performed on a single computer equipped with four Intel®

[6]See https://github.com/pierreganty/mist/wiki.
[7]See http://www.cprover.org/bfc/.
[8]See `backward` at https://github.com/pierreganty/mist/wiki#coverability-checkers-included-in-mist.

Table III. Number of safe instances (top-left), unsafe instances (top-right) and total instances (bottom) decided by every tool. Bold numbers indicate the tool(s) which decide(s) the largest number of instances in the respective category.

Suite	QCover	Petrinizer	mist	bfc	Total
mist	**23**	20	22	20	23
medical	**11**	4	**11**	3	12
bfc	**2**	**2**	**2**	**2**	2
bug_tracking	**32**	**32**	0	19	40
soter	**37**	**37**	0	19	38
Total	**105**	95	35	63	115

Suite	QCover	Petrinizer	mist	bfc	Total
mist	3	—	**4**	**4**	4
medical	—	—	—	—	0
bfc	26	—	29	**42**	44
bug_tracking	0	—	0	**1**	1
soter	8	—	6	**12**	12
Total	37	0	39	**59**	61

Suite	QCover	Petrinizer	mist	bfc	Total
mist	**26**	20	**26**	24	27
medical	**11**	4	**11**	3	12
bfc	28	2	31	**44**	46
bug_tracking	**32**	**32**	0	20	41
soter	**45**	37	6	31	50
Total	**142**	95	74	122	176

Core™ 2.00 GHz i7-4510U CPUs, 8 GB of memory and Ubuntu Linux 14.04 (64 bits). The running time of every tool on an instance was determined using the sum of the `user` and `sys` time reported by the Linux tool `time`.

Table III consists of three tables which display the number of safe instances shown safe, unsafe instances shown unsafe, and the total number of instances of our benchmark suite decided by each individual tool. Our algorithm outperforms all competitors on safe instances, since in this case a proof of safety (i.e. non-coverability) effectively requires the computation of the whole backward coverability set, and this is where pruning via $\mathbb{Q}$-coverability becomes most beneficial. On the other hand, QCover remains competitive on unsafe instances, though a tool such as BFC handles those instances better since its heuristics are more suited for disproving safety (i.e. coverability). Nevertheless, QCover is the overall winner when comparing the number of safe and unsafe instances decided, being far ahead at the top of the leader board deciding 142 out of 176 instances.

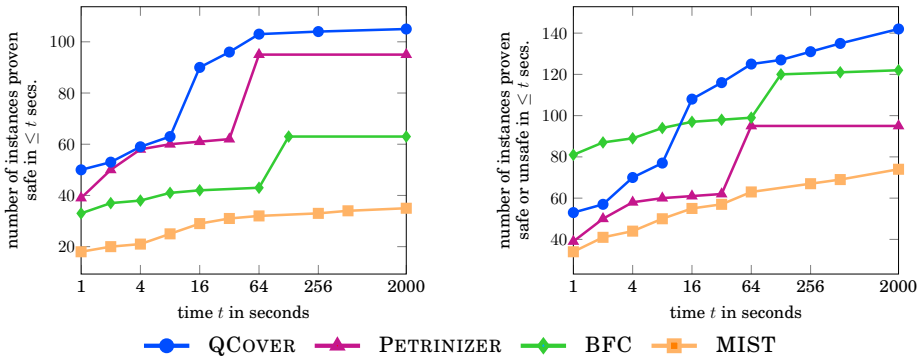

Fig. 2. Cumulative number of instances proven safe (left) and total number of instances decided (right) within a fixed amount of time.

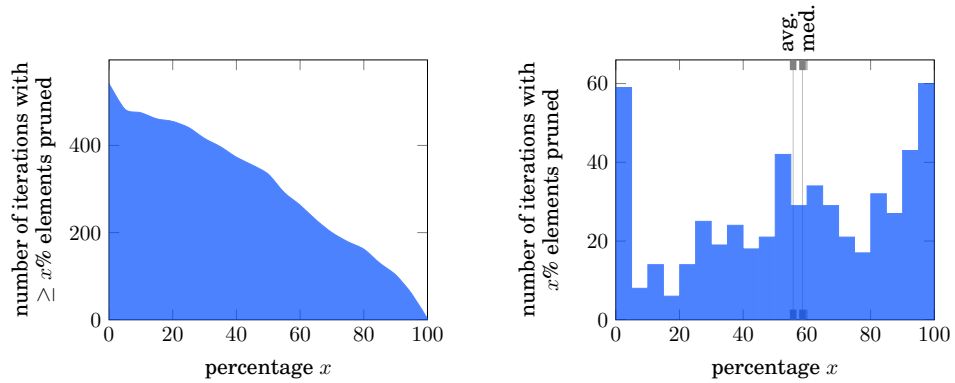

Fig. 3. Number of times a certain percentage of basis elements was removed due to $\mathbb{Q}$-coverability pruning.

QCOVER not only decides more instances, it often does so faster than its competitors. Figure 2 contains two graphs which show the cumulative number of instances proven safe and the total number of instances decided on all suites by each tool within a certain amount of time. When it comes to safety, QCOVER is always ahead of all other tools. However, when looking at all instances decided, BFC first has an advantage. We observed that this advantage occurs on instances of comparably small size. As soon as large instances come into play, QCOVER wins the race. Besides different heuristics used, one reason for this might be the choice of the implementation language (C for BFC vs. PYTHON for QCOVER). In particular, BFC can decide a non-negligible number of instances in less than 10ms, which QCOVER never achieves.

Finally, we consider the effectiveness of using $\mathbb{Q}$-coverability as a pruning criterion. To this end, consider Figure 3 in which we plotted the number of times a certain percentage of basis elements was removed due to not being $\mathbb{Q}$-coverable. Impressively, in some cases more than 95% of the basis elements get discarded. Overall, the average and the median proportion of basis elements discarded are respectively 56% and 59%, which substantiates the usefulness of using $\mathbb{Q}$-coverability as a pruning criterion.

Before we conclude, let us mention that already 83 instances are proven safe by only checking the state equation, and that additionally checking for the Conditions (ii) and (iii) of Proposition 3.1 increases this number to 101 instances. If we use the polynomial-time algorithm of Fraca and Haddad [2015] instead of our FO$\langle\mathbb{Q}, +, <\rangle$ encoding then we can only decide 132 instances in total (within the set time limit). Finally, in our experiments, interpreting variables over $\mathbb{Q}_+$ instead of $\mathbb{N}$ resulted in no measurable overall performance gain.

Concluding, our experimental evaluation demonstrates that the backward algorithm modulo $\mathbb{Q}$-reachability approach to the Petri net coverability problem developed in this paper is highly efficient when run on real-world instances, and superior to existing tools and approaches when compared on standard benchmarks from the literature.

5.4. Relationship to Other Approaches from the Literature

The approach for deciding coverability in Petri nets presented in the previous sections is primarily related to the work by Esparza et al. [2014]; by Kaiser et al. [2014]; and by Delzanno et al. [2001]. Esparza et al. [2014] presented an implementation of a semi-decision procedure for disproving coverability. This semi-decision procedure was originally proposed by Esparza and Melzer [2000] and is based on the Petri-net state equation and traps as sufficient criteria in order to witness non-coverability. As shown

by Esparza and Melzer [2000], those conditions can be encoded into an equi-satisfiable system of linear inequalities called the *trap inequation*. This approach is, however, prone to numerical imprecisions that become problematic even for instances of small size [Esparza and Melzer 2000, Sec. 5.3]. For that reason, Esparza et al. [2014] resort to a CEGAR-based variant of the trap inequation approach which has the drawback that in the worst case, the CEGAR loop has to be executed an exponential number of times leading to an exponential number of queries to the underlying SMT-solver. We will show in Section 5.4.1 that the conditions used by Esparza et al. [2014] are strictly subsumed by a subset of the conditions required to witness coverability in continuous Petri nets: whenever the procedure described therei returns uncoverable then coverability does not hold in the continuous setting either, but not *vice versa*. Thus, a single satisfiability check to our formula in existential $\mathrm{FO}\langle\mathbb{Q}_+,+,<\rangle$ encoding continuous coverability that we developed in this paper completely subsumes the CEGAR-approach presented by Esparza et al. [2014]. Another difference is that we presented a sound *and* complete decision procedure.

Regarding the relationship of our work with the work of Kaiser et al. [2014], they present an approach to coverability in richer classes of well-structured transition systems that is also based on the backward algorithm. They additionally employ a widening heuristic in order to over approximate the minimal basis. Our approach differs in that our minimal basis is always precise yet as small as possible modulo continuous coverability. Thus no backtracking as in the approach of Kaiser et al. [2014] is needed, which is required when the widened basis turns out to be too inaccurate.

The idea of using an over approximation of the reachability set of a Petri net in order to prune minimal basis elements inside the backward algorithm was first described by Delzanno et al. [2001], where place invariants are used as a pruning criterion. However, computing such invariants and checking if a minimal basis element can be pruned potentially requires exponential time.

Finally, a number of further techniques and tools for deciding Petri net coverability or more general well-structured transition systems have been described in the literature. They are, for instance, based on efficient data structures [Ganty 2002; Finkel et al. 2002; Delzanno et al. 2004; Ganty et al. 2007] and generic algorithmic frameworks such as EEC [Geeraerts et al. 2006] and IC3 [Kloos et al. 2013].

5.4.1. Relationship to the Approach of Esparza et al. Esparza et al. [2014] presented a semi-decision procedure for coverability that employs the Petri net state equation and trap constraints inside a CEGAR-framework. Here we discuss in some more detail similarities and differences between our approach and the one of Esparza et al. [2014].

Let $\mathcal{N} = (P, T, \mathbf{Pre}, \mathbf{Post})$ be a Petri net. A *trap in* $\mathcal{N}$ is a non-empty subset of places $Q \subseteq P$ such that $Q^\bullet \subseteq {}^\bullet Q$, and $Q \subseteq P$ is a *siphon in* $\mathcal{N}$ whenever ${}^\bullet Q \subseteq Q^\bullet$. We say that a trap (respectively siphon) is *marked* by a marking $\boldsymbol{m}$ if $\sum_{p\in Q} \boldsymbol{m}(p) > 0$. An important property of marked traps is that they may never become unmarked, i.e., if a trap is marked by some marking $\boldsymbol{m}$, then it will remain marked after any firing sequence starting in $\boldsymbol{m}$. Conversely, when a siphon is unmarked in some marking $\boldsymbol{m}$, it remains so after any firing sequence starting in $\boldsymbol{m}$. By definition, Q is a trap in $\mathcal{N}$ if, and only if, Q is a siphon in $\mathcal{N}^{-1}$.

The coverability criteria that Esparza et al. [2014] build upon are derived from the work of Esparza and Melzer [2000] and can be summarized as follows:

PROPOSITION 5.2 ([ESPARZA ET AL. 2014]). *Let $\mathcal{N} = (P, T, \mathbf{Pre}, \mathbf{Post})$ be a discrete (resp. continuous) Petri net, and $\boldsymbol{m}_0, \boldsymbol{m} \in \mathbb{N}^P$ (resp. $\in \mathbb{Q}_+^P$). If $\boldsymbol{m}_0 \to^* \boldsymbol{m}$ then there exists $\boldsymbol{y} \in \mathbb{N}^T$ (resp. $\in \mathbb{Q}_+^T$) such that*

(i) $\boldsymbol{m} = \boldsymbol{m}_0 + \mathbf{Incid} \cdot \boldsymbol{y}$, *and*

(ii) for every trap $Q \subseteq P$, if Q is marked by $\boldsymbol{m}_0$, then Q is marked by $\boldsymbol{m}$.

In the approach of [Esparza et al. 2014], it is checked whether the conditions of Proposition 5.2 are fulfilled. To this end, the for-all quantifier is replaced by incrementally enumerating all traps in a CEGAR-style fashion. If either condition is violated the semi-decision procedure returns "uncoverable", and "don't know" otherwise. We show that the criteria of Proposition 3.1 imply those of Proposition 5.2:

PROPOSITION 5.3. *Let $\mathcal{N} = (P, T, \mathbf{Pre}, \mathbf{Post})$ be a continuous Petri net, and let $\boldsymbol{m}_0, \boldsymbol{m} \in \mathbb{Q}_+^P$. If $\mathcal{N}, \boldsymbol{m}_0, \boldsymbol{m}$ satisfy Conditions (i) and (iii) of Proposition 3.1, then they also satisfy Conditions (i) and (ii) of Proposition 5.2.*

PROOF. We proceed by contraposition, i.e., we show that if for every $\boldsymbol{y} \in \mathbb{Q}_+^T$, one of the conditions of Proposition 5.2 is not satisfied, then for every $\boldsymbol{y} \in \mathbb{Q}_+^T$, one of conditions (i) or (iii) of Proposition 3.1 is not satisfied.

Let $\boldsymbol{y} \in \mathbb{Q}_+^T$. Suppose that $\boldsymbol{y}$ does not satisfy Condition (i) of Proposition 5.2. Since this condition is identical to Condition (i) of Proposition 3.1, we are done. Therefore, we may assume that $\boldsymbol{y}$ satisfies Condition (i) of Proposition 5.2, i.e.,

$$\boldsymbol{m} = \boldsymbol{m}_0 + \mathbf{Incid} \cdot \boldsymbol{y} \ , \tag{4}$$

but not its Condition (ii). Thus, there exists a trap $Q \subseteq P$ in $\mathcal{N}$ marked by $\boldsymbol{m}_0$, but not marked by $\boldsymbol{m}$. We note that Q is a siphon in $\mathcal{N}^{-1}$ marked by $\boldsymbol{m}_0$, but not marked by $\boldsymbol{m}$. Let $T' \stackrel{\text{def}}{=} [\![\boldsymbol{y}]\!]$, $P' \stackrel{\text{def}}{=} {}^\bullet T'^\bullet$ and $Q' \stackrel{\text{def}}{=} Q \cap P'$. We claim that

$$Q' \text{ is a siphon in } \mathcal{N}_{T'}^{-1} \text{ that is not marked by } \boldsymbol{m}. \tag{5}$$

From this claim, we are done. Indeed, by [Fraca and Haddad 2015, Prop. 18], Claim (5) implies that $T' \notin fs(\mathcal{N}_{T'}^{-1}, \boldsymbol{m})$. Consequently, $[\![\boldsymbol{y}]\!] \notin fs(\mathcal{N}^{-1}, \boldsymbol{m})$, hence Condition (iii) of Proposition 3.1 is not satisfied.

Let us prove Claim (5). Since Q is not marked by $\boldsymbol{m}$, $Q' \subseteq Q$ is also not marked by $\boldsymbol{m}$. Let $t \in T'$ be such that $t \in {}^\bullet Q'$. Since $Q' \subseteq Q$, we also have that $t \in {}^\bullet Q$. Moreover, since Q is siphon in $\mathcal{N}^{-1}$, we have that $t \in Q^\bullet$. By definition, $t \in P'^\bullet$, hence $t \in Q^\bullet \cap P'^\bullet$ and consequently $t \in Q'^\bullet$. Therefore, ${}^\bullet Q' \subseteq Q'^\bullet$. In order to show that Q' is indeed a siphon, it remains to show that $Q' \neq \emptyset$. Since Q is marked by $\boldsymbol{m}_0$, but not by $\boldsymbol{m}$, there exists $p \in Q$ such that $\boldsymbol{m}_0(p) > 0$ and $\boldsymbol{m}(p) = 0$. By (4), $\boldsymbol{m} = \boldsymbol{m}_0 + \mathbf{Incid} \cdot \boldsymbol{y}$. Thus,

$$\begin{aligned}
0 &= \boldsymbol{m}_0(p) + (\mathbf{Incid} \cdot \boldsymbol{y})(p) \\
&= \boldsymbol{m}_0(p) + \sum_{t \in T} \mathbf{Incid}(p,t) \cdot \boldsymbol{y}(t) \\
&= \boldsymbol{m}_0(p) + \sum_{t \in [\![\boldsymbol{y}]\!]} \mathbf{Incid}(p,t) \cdot \boldsymbol{y}(t) && \text{(since } \boldsymbol{y}(t) = 0 \text{ for } t \notin [\![\boldsymbol{y}]\!]) \\
&= \boldsymbol{m}_0(p) + \sum_{t \in T'} \mathbf{Incid}(p,t) \cdot \boldsymbol{y}(t) && \text{(by } T' = [\![\boldsymbol{y}]\!]).
\end{aligned}$$

Since $\boldsymbol{m}_0(p) > 0$, there exists some $t \in T'$ such that $\mathbf{Incid}(p,t) < 0$. Therefore, $p \in {}^\bullet t^\bullet$, whence $p \in P'$. Consequently $p \in Q \cap P' = Q'$, which concludes the proof. □

In fact, we may strengthen the previous proposition by showing that Proposition 3.1 is stronger than Proposition 5.2:

PROPOSITION 5.4. *There exists a continuous Petri net $\mathcal{N} = (P, T, \mathbf{Pre}, \mathbf{Post})$, and markings $\boldsymbol{m}_0, \boldsymbol{m} \in \mathbb{Q}_+^P$ satisfying the conditions of Proposition 5.2, but not satisfying the Conditions (i) and (iii) of Proposition 3.1.*

PROOF. Let $\mathcal{N} = (\{p, q\}, \{s, t\}, \mathbf{Pre}, \mathbf{Post})$ be the continuous Petri net depicted in Figure 4, $\boldsymbol{m}_0 = (1, 0)$ and $\boldsymbol{m} = (0, 1)$. We note that $\boldsymbol{m}$ is not reachable from $\boldsymbol{m}_0$. Indeed,

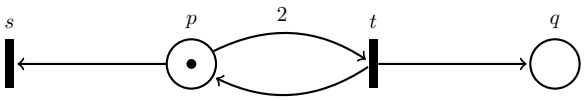

Fig. 4. Example of continuous Petri net for which the conditions of Proposition 5.2 are not sufficient to certify non-reachability.

the unique solution to $\boldsymbol{m} = \boldsymbol{m}_0 + \mathbf{Incid} \cdot \boldsymbol{y}$ is $\boldsymbol{y} = (0, 1)$, yet it is impossible to fully fire t in $\boldsymbol{m}_0$. The unique trap of $\mathcal{N}$ is $\{q\}$ and it is not marked by $\boldsymbol{m}_0$. Therefore, $\mathcal{N}, \boldsymbol{m}_0$ and $\boldsymbol{m}$ satisfy conditions of Proposition 5.2.

Suppose that Conditions (i) and (iii) of Proposition 3.1 are satisfied. They must be satisfied by the unique solution $\boldsymbol{y} = (0, 1)$. Thus, $\{t\} = [\![\boldsymbol{y}]\!] \in fs_{\mathcal{N}^{-1}}(\boldsymbol{m})$. By [Fraca and Haddad 2015, Prop. 18], $\mathcal{N}^{-1}_{\{t\}}$ does not possess any siphon marked by $\boldsymbol{m}$. Yet, $\{q\}$ is a siphon in $\mathcal{N}^{-1}_{\{t\}}$ marked by $\boldsymbol{m}$. This is contradiction, hence Conditions (i) and (iii) of Proposition 3.1 are not satisfied. □

The previous proposition shows that the single formula stated in Proposition 3.2 strictly subsumes the approach of Esparza et al. [2014]. Moreover, it provides a theoretical justification for why the approach of Esparza et al. [2014] performs so well in practice: the conditions are a strict subset of the conditions developed by Fraca and Haddad [2015] for $\mathbb{Q}$-reachability.

6. CONCLUSION

In this paper, we developed a characterization of the reachability relation for continuous Petri nets in existential $\mathrm{FO}\langle\mathbb{Q}, +, <\rangle$. Given a Petri net $\mathcal{N}$, we showed how to compute in linear time a formula $\Phi_{\mathcal{N}}(\boldsymbol{x}, \boldsymbol{x}')$ whose set of solutions defines the continuous reachability relation of $\mathcal{N}$. Using this characterization as a starting point, we derived novel upper bounds for standard decision problem for continuous Petri nets, such as inclusion, (structural) ε-liveness, and the (existential) home state problem. Moreover, we showed how to integrate continuous coverability checks as a pruning heuristic inside the backward algorithm for deciding coverability in discrete Petri nets. In particular, the logical characterization of continuous reachability enables the use of SMT solvers in order to decide continuous coverability, and in effect we obtained a decision procedure for the Petri net coverability problem that outperforms all its competitors on standard benchmarks from the literature. At the time of writing, Geffroy et al. [2016] have adjusted our encoding of continuous reachability to specifically target continuous coverability, and reported a two-fold speed-up on the benchmarks used in this paper.

Using pruning invariants inside the backward algorithm is not a novelty as such and was first described by Delzanno et al. [2001]. However, we believe that this paper demonstrates that the significant progress on SMT solvers that has taken place over the last fifteen years provides new perspectives on developing and applying invariants which are definable in a logical theory that an SMT solver can handle. In particular, over approximations of reachability sets via arithmetic theories have been developed for even more expressive models, for instance for Petri nets with resets [Chistikov et al. 2017] for which coverability is Ackermann-complete [Schnoebelen 2010], and could directly be integrated inside the backward algorithm as proposed in this paper (provided that those more expressive models are well structured). We believe that this approach will enable the practical algorithmic analysis of models that have mostly

been studied from a theoretical perspective due to the high worst-case complexity of their decision problems.

ACKNOWLEDGMENTS

We would like to thank Vincent Antaki for an early implementation of the polynomial-time algorithm of Fraca and Haddad [2015], and also Gilles Geeraerts for his support with the MIST file format. We are grateful to the anonymous reviewers of TACAS'16 and ACM TOCL for their comments and suggestions.

REFERENCES

Parosh Aziz Abdulla, Karlis Cerans, Bengt Jonsson, and Yih-Kuen Tsay. 1996. General Decidability Theorems for Infinite-State Systems. In *Logic in Computer Science, LICS*. IEEE Computer Society, 313–321. DOI:http://dx.doi.org/10.1109/LICS.1996.561359

André Arnold and Michel Latteux. 1978. Récursivit et cônes rationnels fermés par intersection. *Calcolo* 15, 4 (1978), 381–394. DOI:http://dx.doi.org/10.1007/BF02576519

Thomas Ball, Sagar Chaki, and Sriram K. Rajamani. 2001. Parameterized Verification of Multithreaded Software Libraries. In *Tools and Algorithms for the Construction and Analysis of Systems, TACAS (Lect. Notes Comp. Sci.)*, Vol. 2031. Springer, 158–173. DOI:http://dx.doi.org/10.1007/3-540-45319-9_12

Leonard Berman. 1980. The Complexitiy of Logical Theories. *Theor. Comput. Sci.* 11 (1980), 71–77. DOI:http://dx.doi.org/10.1016/0304-3975(80)90037-7

Eike Best and Javier Esparza. 2016. Existence of home states in Petri nets is decidable. *Inf. Process. Lett.* 116, 6 (2016), 423–427. DOI:http://dx.doi.org/10.1016/j.ipl.2016.01.011

Michael Blondin, Alain Finkel, Christoph Haase, and Serge Haddad. 2016. Approaching the Coverability Problem Continuously. In *Tools and Algorithms for Construction and Analysis of Systems, TACAS (Lect. Notes Comp. Sci.)*, Vol. 9636. Springer, 480–496. DOI:http://dx.doi.org/10.1007/978-3-662-49674-9_28

Itshak Borosh and Leon B. Treybing. 1976. Bounds on Positive Integral Solutions of Linear Diophantine Equations. *P. Am. Math. Soc.* 55, 2 (1976), 299–304. DOI:http://dx.doi.org/10.2307/2041711

Laura Bozzelli and Pierre Ganty. 2011. Complexity Analysis of the Backward Coverability Algorithm for VASS. In *Reachability Problems, RP (Lect. Notes Comp. Sci.)*, Vol. 6945. Springer, 96–109. DOI:http://dx.doi.org/10.1007/978-3-642-24288-5_10

E. Cardoza, Richard J. Lipton, and Albert R. Meyer. 1976. Exponential Space Complete Problems for Petri Nets and Commutative Semigroups: Preliminary Report. In *Symposium on Theory of Computing, STOC*. 50–54. DOI:http://dx.doi.org/10.1145/800113.803630

Dmitry Chistikov, Christoph Haase, and Simon Halfon. 2017. Context-free commutative grammars with integer counters and resets. *Theoret. Comput. Sci.* (2017), –. DOI:http://dx.doi.org/10.1016/j.tcs.2016.06.017 In press.

René David and Hassane Alla. 1987. Continuous Petri nets. In *Proc. 8th European Workshop on Application and Theory of Petri nets*. 275–294.

René David and Hassane Alla. 2010. *Discrete, Continuous, and Hybrid Petri nets* (2nd ed.). Springer. DOI:http://dx.doi.org/10.1007/978-3-642-10669-9

Davide de Frutos Escrig and Colette Johnen. 1989. *Decidability of home space property*. Technical Report LRI-503. Univ. de Paris-Sud, Centre d'Orsay, Laboratoire de Recherche en Informatique.

Leonardo Mendonça de Moura and Nikolaj Bjørner. 2008. Z3: An Efficient SMT Solver. In *Tools and Algorithms for the Construction and Analysis of Systems, TACAS (Lect. Notes Comp. Sci.)*, Vol. 4963. Springer, 337–340. DOI:http://dx.doi.org/10.1007/978-3-540-78800-3_24

Giorgio Delzanno, Jean-François Raskin, and Laurent Van Begin. 2001. Attacking Symbolic State Explosion. In *Computer Aided Verification, CAV (Lect. Notes Comp. Sci.)*, Vol. 2102. Springer, 298–310. DOI:http://dx.doi.org/10.1007/3-540-44585-4_28

Giorgio Delzanno, Jean-François Raskin, and Laurent Van Begin. 2004. Covering sharing trees: a compact data structure for parameterized verification. *Int. J. Softw. Tools Technol. Transf.* 5, 2-3 (2004), 268–297. DOI:http://dx.doi.org/10.1007/s10009-003-0110-0

Jörg Desel and Javier Esparza. 1995. *Free Choice Petri nets*. Cambridge Tracts in Theoret. Comp. Sci., Vol. 40. Cambridge University Press, New York, NY, USA. DOI:http://dx.doi.org/10.1017/CBO9780511526558

Emanuele D'Osualdo, Jonathan Kochems, and C.-H. Luke Ong. 2013. Automatic Verification of Erlang-Style Concurrency. In *Static Analysis, SAS (Lect. Notes Comp. Sci.)*, Vol. 7935. Springer, 454–476. DOI:http://dx.doi.org/10.1007/978-3-642-38856-9_24

Emanuele D'Osualdo, Jonathan Kochems, and Luke Ong. 2012. Soter: an automatic safety verifier for Erlang. In *Programming based on Actors, Agents, and Decentralized Control, AGERE*. ACM, 137–140. DOI:http://dx.doi.org/10.1145/2414639.2414658

Javier Esparza, Ruslán Ledesma-Garza, Rupak Majumdar, Philipp Meyer, and Filip Nikšić. 2014. An SMT-Based Approach to Coverability Analysis. In *Computer Aided Verification, CAV (Lect. Notes Comp. Sci.)*, Vol. 8559. Springer, 603–619. DOI:http://dx.doi.org/10.1007/978-3-319-08867-9_40

Javier Esparza and Stephan Melzer. 2000. Verification of Safety Properties Using Integer Programming: Beyond the State Equation. *Formal Methods in System Design* 16, 2 (2000), 159–189. DOI:http://dx.doi.org/10.1023/A:1008743212620

Alain Finkel and Jérôme Leroux. 2015. Recent and simple algorithms for Petri nets. *Software and System Modeling* 14, 2 (2015), 719–725. DOI:http://dx.doi.org/10.1007/s10270-014-0426-0

Alain Finkel, Jean-François Raskin, Mathias Samuelides, and Laurent Van Begin. 2002. Monotonic Extensions of Petri Nets: Forward and Backward Search Revisited. *Electr. Notes Theor. Comput. Sci.* 68, 6 (2002), 85–106. DOI:http://dx.doi.org/10.1016/S1571-0661(04)80535-8

Alain Finkel and Philippe Schnoebelen. 2001. Well-structured transition systems everywhere! *Theor. Comput. Sci.* 256, 1-2 (2001), 63–92. DOI:http://dx.doi.org/10.1016/S0304-3975(00)00102-X

Estíbaliz Fraca and Serge Haddad. 2015. Complexity Analysis of Continuous Petri Nets. *Fundam. Inform.* 137, 1 (2015), 1–28. DOI:http://dx.doi.org/10.3233/FI-2015-1168

Pierre Ganty. 2002. *Algorithmes et structures de données efficaces pour la manipulation de contraintes sur les intervalles (in French)*. Master's thesis. Université Libre de Bruxelles, Belgium.

Pierre Ganty, Cédric Meuter, Giorgio Delzanno, Gabriel Kalyon, Jean-François Raskin, and Laurent Van Begin. 2007. *Symbolic Data Structure for sets of k-uples*. Technical Report 570. Université Libre de Bruxelles, Belgium.

Gilles Geeraerts, Jean-François Raskin, and Laurent Van Begin. 2006. Expand, Enlarge and Check: New algorithms for the coverability problem of WSTS. *J. Comput. Syst. Sci.* 72, 1 (2006), 180–203. DOI:http://dx.doi.org/10.1016/j.jcss.2005.09.001

Gilles Geeraerts, Jean-François Raskin, and Laurent Van Begin. 2010. On the Efficient Computation of the Minimal Coverability Set of Petri Nets. *Int. J. Found. Comput. Sci.* 21, 2 (2010), 135–165. DOI:http://dx.doi.org/10.1142/S0129054110007180

Thomas Geffroy, Jérôme Leroux, and Grégoire Sutre. 2016. Occam's Razor Applied to the Petri Net Coverability Problem. In *Reachability Problems, RP (Lect. Notes Comp. Sci.)*, Vol. 9899. Springer, 77–89. DOI:http://dx.doi.org/10.1007/978-3-319-45994-3_6

Steven M. German and A. Prasad Sistla. 1992. Reasoning about Systems with Many Processes. *J. ACM* 39, 3 (1992), 675–735. DOI:http://dx.doi.org/10.1145/146637.146681

Michel Henri Théodore Hack. 1974. The Recursive Equivalence of the Reachability Problem and the Liveness Problem for Petri Nets and Vector Addition Systems. In *Switching and Automata Theory, SWAT (FOCS)*. IEEE Computer Society, 156–164. DOI:http://dx.doi.org/10.1109/SWAT.1974.28

Michel Henri Théodore Hack. 1976. *Decidability questions for Petri Nets*. Ph.D. Dissertation. Massachusetts Institute of Technology, USA.

Monika Heiner, David R. Gilbert, and Robin Donaldson. 2008. Petri Nets for Systems and Synthetic Biology. In *Formal Methods for Computational Systems Biology, 8th International School on Formal Methods for the Design of Computer, Communication, and Software Systems, SFM (Lect. Notes Comp. Sci.)*, Vol. 5016. Springer, 215–264. DOI:http://dx.doi.org/10.1007/978-3-540-68894-5_7

Petr Jančar. 2017. Deciding Structural Liveness of Petri Nets. In *Current Trends in Theory and Practice of Computer Science, SOFSEM (Lect. Notes Comp. Sci.)*, Vol. 10139. Springer, 91–102. DOI:http://dx.doi.org/10.1007/978-3-319-51963-0_8

Ji Jank. 2009. *Issue Tracking Systems*. Master's thesis. Masarykova univerzita, Czech Republic.

Alexander Kaiser, Daniel Kroening, and Thomas Wahl. 2012. Efficient Coverability Analysis by Proof Minimization. In *Concurrency Theory, CONCUR*. Springer, 500–515. DOI:http://dx.doi.org/10.1007/978-3-642-32940-1_35

Alexander Kaiser, Daniel Kroening, and Thomas Wahl. 2014. A Widening Approach to Multithreaded Program Verification. *ACM Trans. Program. Lang. Syst.* 36, 4 (2014), 14:1–14:29. http://doi.acm.org/10.1145/2629608

Richard M. Karp and Raymond E. Miller. 1967. Parallel Program Schemata: A Mathematical Model for Parallel Computation. In *Switching and Automata Theory, SWAT (FOCS)*. IEEE Computer Society, 55–61. DOI:http://dx.doi.org/10.1109/FOCS.1967.27

Johannes Kloos, Rupak Majumdar, Filip Niksic, and Ruzica Piskac. 2013. Incremental, Inductive Coverability. In *Computer Aided Verification, CAV (Lect. Notes Comp. Sci.)*, Vol. 8044. Springer, 158–173. DOI:http://dx.doi.org/10.1007/978-3-642-39799-8_10

S. Rao Kosaraju. 1982. Decidability of Reachability in Vector Addition Systems (Preliminary Version). In *Symposium on Theory of Computing, STOC*. ACM, 267–281. DOI:http://dx.doi.org/10.1145/800070.802201

Jean-Luc Lambert. 1992. A Structure to Decide Reachability in Petri Nets. *Theoret. Comput. Sci.* 99, 1 (1992), 79–104. DOI:http://dx.doi.org/10.1016/0304-3975(92)90173-D

Jérôme Leroux. 2009. The General Vector Addition System Reachability Problem by Presburger Inductive Invariants. In *Logic in Computer Science, LICS*. IEEE Computer Society, 4–13. DOI:http://dx.doi.org/10.1109/LICS.2009.10

Jérôme Leroux. 2011. Vector Addition System Reachability Problem: a Short Self-contained Proof. In *Language and Automata Theory and Applications, LATA (Lect. Notes Comp. Sci.)*, Vol. 6638. Springer, 41–64. DOI:http://dx.doi.org/10.1007/978-3-642-21254-3_3

Jérôme Leroux. 2012. Vector Addition Systems Reachability Problem (A Simpler Solution). In *The Alan Turing Centenary Conference*. EasyChair, 214–228. http://www.easychair.org/publications/?page=1673703727

Jérôme Leroux and Sylvain Schmitz. 2015. Demystifying Reachability in Vector Addition Systems. In *Logic in Computer Science, LICS*. IEEE, 56–67. DOI:http://dx.doi.org/10.1109/LICS.2015.16

Richard J. Lipton. 1976. *The Reachability Problem Requires Exponential Space*. Technical Report 63. Department of Computer Science, Yale University.

Rupak Majumdar, Roland Meyer, and Zilong Wang. 2013. Static Provenance Verification for Message Passing Programs. In *Static Analysis, SAS (Lect. Notes Comp. Sci.)*, Vol. 7935. Springer, 366–387. DOI:http://dx.doi.org/10.1007/978-3-642-38856-9_20

Ernst W. Mayr. 1981. An Algorithm for the General Petri Net Reachability Problem. In *Symposium on Theory of Computing, STOC*. ACM, 238–246. DOI:http://dx.doi.org/10.1145/800076.802477

Charles Rackoff. 1978. The Covering and Boundedness Problems for Vector Addition Systems. *Theor. Comput. Sci.* 6 (1978), 223–231. DOI:http://dx.doi.org/10.1016/0304-3975(78)90036-1

Laura Recalde, Enrique Teruel, and Manuel Silva Suárez. 1999. Autonomous Continuous P/T Systems. In *Application and Theory of Petri Nets, ICATPN (Lect. Notes Comp. Sci.)*, Vol. 1639. Springer, 107–126. DOI:http://dx.doi.org/10.1007/3-540-48745-X_8

Venkatramana N. Reddy, Michael N. Liebman, and Michael L. Mavrovouniotis. 1996. Qualitative analysis of biochemical reaction systems. *Comput. Biol. Med.* 26, 1 (1996), 9–24. DOI:http://dx.doi.org/10.1016/0010-4825(95)00042-9

Pierre-Alain Reynier and Frédéric Servais. 2013. Minimal Coverability Set for Petri Nets: Karp and Miller Algorithm with Pruning. *Fundam. Inform.* 122, 1-2 (2013), 1–30. DOI:http://dx.doi.org/10.3233/FI-2013-781

Philippe Schnoebelen. 2010. Revisiting Ackermann-Hardness for Lossy Counter Machines and Reset Petri Nets. In *Mathematical Foundations of Computer Science, MFCS (Lect. Notes Comp. Sci.)*, Vol. 6281. Springer, 616–628. DOI:http://dx.doi.org/10.1007/978-3-642-15155-2_54

Alexander Schrijver. 1998. *Theory of Linear and Integer Programming*. Wiley.

Eduardo D. Sontag. 1985. Real Addition and the Polynomial Hierarchy. *Inf. Process. Lett.* 20, 3 (1985), 115–120. DOI:http://dx.doi.org/10.1016/0020-0190(85)90076-6

Filip Thoen and Francky Catthoor. 2000. *Modeling, verification and exploration of task-level concurrency in real-time embedded systems*. Springer US, Boston, MA.

Antti Valmari and Henri Hansen. 2014. Old and New Algorithms for Minimal Coverability Sets. *Fundam. Inform.* 131, 1 (2014), 1–25. DOI:http://dx.doi.org/10.3233/FI-2014-1002

Wil M.P. van der Aalst. 1998. The application of Petri nets to workflow management. *J. Circuit. Syst. Comp.* 8, 1 (1998), 21–66. DOI:http://dx.doi.org/10.1142/S0218126698000043

Kumar Neeraj Verma, Helmut Seidl, and Thomas Schwentick. 2005. On the Complexity of Equational Horn Clauses. In *Automated Deduction - CADE-20 (Lect. Notes Comp. Sci.)*, Vol. 3632. Springer, 337–352. DOI:http://dx.doi.org/10.1007/11532231_25